INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

N° attribué par la bibliothèque
|__|__|__|__|__|__|__|__|__|__|

T H È S E

pour obtenir le grade de

DOCTEUR DE L'INPG

Spécialité : IMAGERIE, VISION ET ROBOTIQUE

préparée au laboratoire GRAVIR
Ecole Doctorale MATHÉMATIQUES, INFORMATIQUE, SCIENCES ET TECHNOLOGIES DE L'INFORMATION

présentée et soutenue publiquement par :

Adrien Bartoli

le 10 septembre 2003

Titre :

Reconstruction et alignement en vision 3D : points, droites, plans et caméras

Directeurs de thèse :
Peter Sturm et Radu Horaud

Jury :

James Crowley	Président
Michel Dhome	Rapporteur
Nassir Navab	Rapporteur
Radu Horaud	Directeur de thèse
Peter Sturm	Co-encadrant
Ian Reid	Examinateur

Résumé

Cette thèse concerne la reconstruction de modèles 3D de scènes à partir d'images prises par des caméras. Il est courant de reconstruire à partir de sous-ensembles d'images puis de fusionner les modèles partiels ainsi obtenus par une phase d'alignement 3D. Les algorithmes de reconstruction et d'alignement s'appuient sur des correspondances de points ou de droites entre les images. La localisation de ces points ou droites dans les images est affectée par un bruit de mesure, influençant la qualité des modèles 3D reconstruits. Cette thèse est centrée sur l'obtention de résultats optimaux et sur les problèmes de représentation qui en découlent.

La première partie de cette thèse aborde le problème de la reconstruction de modèles 3D. Les cas des caméras calibrées et non calibrées sont traités. Nous développons des méthodes de reconstruction de points, de droites et de caméras. L'incorporation de contraintes géométriques de coplanarité permet la reconstruction conjointe de plans. Nos contributions principales sont le développement et la comparaison de méthodes permettant la reconstruction 3D optimale de points, droites, plans et caméras.

La deuxième partie de cette thèse aborde le problème de l'alignement de modèles 3D, qui consiste à estimer la transformation géométrique liant deux modèles 3D. Les méthodes existantes sont basées sur des correspondances de points. Nous étudions le cas des correspondances de droites. Les cas des caméras calibrées et non calibrées sont traités. Nos contributions majeures dans ce domaine sont, d'un point de vue théorique, une étude des transformations géométriques de droites 3D. Plus précisément, nous étendons la représentation matricielle standard, adaptée aux points, en une représentation adaptée aux droites. D'un point de vue pratique, nous développons et comparons plusieurs méthodes d'alignement linéaires et non-linéaires.

Nous proposons finalement des méthodes de reconstruction de modèles 3D lorsque la scène observée n'est pas rigide. Par ailleurs, nous développons un méthode de détection automatique de surfaces planes dans une modèle 3D.

Mots clés : vision par ordinateur, reconstruction tridimensionnelle, reconstruction projective, ajustement de faisceaux.

Abstract

This thesis deals with the reconstruction of 3D models of scenes from images taken by cameras. It is common to reconstruct from subsets of images and to merge the partial 3D models, based on a 3D alignment step. Reconstruction and alignment algorithms rely on point or line correspondences between the images. The position of these features in the images is affected by noise, which influences the quality of the reconstructed 3D models. This thesis focuses on obtaining optimal results and on the problems of representation which are induced.

The first part of this thesis tackles the problem of reconstructing 3D models. Calibrated and uncalibrated cameras are dealt with. We develop methods for the reconstruction of points, lines and cameras. The incorporation of geometric coplanarity constraints allows simultaneous reconstruction of planes. Our main contributions are the development and the comparison of methods for the optimal 3D reconstruction of points, lines, planes and cameras.

The second part of this thesis is concerned with the problem of aligning 3D models, or equivalently, estimating the geometric transformation relating two 3D models. Existing methods are based on point correspondences. We study the case of line correspondences. Calibrated and uncalibrated cameras are dealt with. Our major contributions in this domain are, from a theoretical point of view, the study of geometric transformations of 3D lines. More precisely, we extend the standard matrix representation, suitable for points, to a representation suitable for lines. From a practical point of view, we develop and compare numerous linear and non-linear alignment methods.

Finally, we propose methods for the reconstruction of 3D models for non-rigid scenes and a method for the automatic detection of planar surfaces within a 3D model.

Keywords : computer vision, threedimensional reconstruction, projective reconstruction, bundle adjustment.

Table des matières

1 Introduction **1**

1.1 Modélisation et données d'entrée . . . 2
1.2 Stratégies et tâches de reconstruction . . . 2
1.3 Reconstruction optimale . . . 3
1.3.1 Initialisation . . . 3
1.3.2 Paramétrisation . . . 4
1.4 Contributions . . . 4
1.4.1 Lignes directrices . . . 4
1.4.2 Paramétrisations et algorithmes . . . 5

2 Eléments de base **7**

2.1 Conventions et notations . . . 8
2.2 Espaces Euclidien, métrique, affine et projectif . . . 9
2.3 La matrice des cofacteurs . . . 9
2.4 Quelques décompositions de matrices . . . 11
2.4.1 Décomposition en valeurs singulières . . . 11
2.4.2 Décomposition QR . . . 11
2.5 Mesures de distance . . . 12
2.5.1 Distances Euclidiennes . . . 12
2.5.2 Distances algébriques . . . 13
2.5.3 Relier les distances Euclidiennes et algébriques . . . 14
2.6 Estimation d'entités géométriques, optimisation linéaire et quasi-linéaire . . . 15
2.6.1 Optimisation linéaire . . . 15
2.6.2 Optimisation quasi-linéaire . . . 17
2.6.3 Systèmes non homogènes . . . 18
2.6.4 Optimisation sous contraintes linéaires . . . 19
2.7 Matrices orthonormales et de rotation . . . 20
2.7.1 Définition . . . 20
2.7.2 Décomposition par angles élémentaires des matrices de rotation . . . 21
2.7.3 Estimation des matrices de rotation par mise à jour locale minimale . . . 21
2.8 Le modèle de caméra perspectif . . . 23
2.9 Stratification de la reconstruction et du calibrage, reconstructabilité . . . 25
2.10 Géométrie d'un système de caméras non calibrées . . . 26
2.10.1 Le cas de deux vues : la géométrie épipolaire . . . 26
2.10.2 Le cas multi-vues . . . 29
2.11 Observation d'un plan . . . 29
2.12 Les coordonnées de Plücker . . . 31

3 Cadre général de la reconstruction par ajustement de faisceaux 35
3.1 Introduction . . . 36
3.2 Initialisation . . . 37
3.2.1 Méthodes directes . . . 37
3.2.2 Méthodes séquentielles . . . 39
3.3 L'erreur de reprojection . . . 39
3.3.1 Le cas des points . . . 39
3.3.2 Le cas des droites . . . 40
3.3.3 L'erreur de reprojection et l'erreur d'estimation . . . 42
3.4 Libertés de jauge . . . 42
3.5 Paramétrisation . . . 44
3.5.1 Motivations et fonction de paramétrisation . . . 44
3.5.2 Mise à jour . . . 44
3.5.3 Influence des libertés de jauge . . . 45
3.5.4 Evaluer la qualité d'une paramétrisation . . . 46
3.5.5 Exemple de paramétrisation . . . 46
3.6 Optimisation non-linéaire . . . 47
3.6.1 L'algorithme de Gauss-Newton . . . 47
3.6.2 L'algorithme de Levenberg-Marquardt . . . 48
3.6.3 Structure et schéma de résolution éparse des équations normales . . . 48
3.6.4 Reconstruction avec contraintes géométriques . . . 49
3.7 Conclusions . . . 50

4 Géométrie de deux caméras non calibrées et estimation non-linéaire de la matrice fondamentale 51
4.1 Introduction . . . 52
4.2 Factorisation de la matrice fondamentale . . . 53
4.2.1 Factorisation partielle . . . 54
4.2.2 Factorisation totale . . . 56
4.2.3 Résumé . . . 58
4.3 Paramétrisation de la matrice fondamentale . . . 58
4.3.1 La paramétrisation de Luong et Faugeras . . . 59
4.3.2 Les paramétrisations de Zhang et de Csurka et al. . . . 61
4.3.3 La solution de Zhang et Loop par transformation projective des images . . . 63
4.3.4 La solution de Torr et Zisserman par correspondances de points . . . 63
4.3.5 La paramétrisation "épipôle + homographie" . . . 63
4.4 La représentation orthonormale de la matrice fondamentale . . . 63
4.4.1 Décomposition . . . 64
4.4.2 Recomposition . . . 64
4.4.3 Consistance et complétude . . . 65
4.4.4 Réalisation . . . 65
4.4.5 Optimisation minimale . . . 65
4.4.6 L'approche réduite : minimisation des distances point-droite épipolaire . . . 70
4.4.7 L'approche directe : minimisation de l'erreur de reprojection . . . 71
4.4.8 Matrices essentielles . . . 72
4.5 Résultats expérimentaux . . . 73
4.5.1 Données simulées . . . 74
4.5.2 Données réelles . . . 76
4.6 Conclusions et perspectives . . . 76

5 Reconstruction de plans, de points et de caméras **79**
5.1 Introduction . . . 80
5.2 Modélisation du problème . . . 82
5.2.1 Gestion des contraintes de multi-coplanarité . . . 82
5.2.2 Etude du nombre de degrés de liberté . . . 83
5.2.3 Reconstructabilité . . . 84
5.3 Etat de l'art et approche proposée . . . 84
5.3.1 Initialisation . . . 84
5.3.2 Ajustement de faisceaux plan par morceaux . . . 87
5.4 Le cas de deux vues . . . 89
5.4.1 Initialisation . . . 89
5.4.2 Ajustement de faisceaux plan par morceaux . . . 95
5.5 Le cas multi-vues . . . 97
5.5.1 Initialisation . . . 98
5.5.2 Ajustement de faisceaux plan par morceaux . . . 102
5.6 Autres types de contraintes . . . 105
5.7 Résultats expérimentaux . . . 106
5.7.1 Données simulées . . . 106
5.7.2 Données réelles . . . 110
5.8 Conclusions et perspectives . . . 118

6 Reconstruction de droites et de caméras **121**
6.1 Introduction . . . 122
6.2 Représentation des droites 3D . . . 124
6.2.1 Représentations complètes . . . 124
6.2.2 Représentations partielles . . . 125
6.2.3 Résumé . . . 126
6.3 Triangulation . . . 126
6.3.1 Travaux précédents . . . 127
6.3.2 Choix de la représentation . . . 128
6.3.3 Projection d'une droite en coordonnées de Plücker . . . 128
6.3.4 Algorithme linéaire . . . 129
6.3.5 Correction de Plücker . . . 130
6.3.6 Algorithme quasi-linéaire 1 . . . 130
6.3.7 Algorithme quasi-linéaire 2 . . . 131
6.3.8 Algorithme non-linéaire . . . 132
6.4 Ajustement de faisceaux . . . 133
6.4.1 Etat de l'art . . . 133
6.4.2 La représentation orthonormale des droites 3D . . . 133
6.5 Résultats expérimentaux . . . 141
6.5.1 Données simulées . . . 142
6.5.2 Données réelles . . . 143
6.6 Conclusions et perspectives . . . 149

7 Les matrices de mouvement pour droites 3D **155**
7.1 Introduction . . . 156
7.2 Les matrices de mouvement pour droites 3D . . . 157
7.2.1 Cas générique . . . 157
7.2.2 Cas projectif . . . 160
7.2.3 Cas affine . . . 162
7.2.4 Cas métrique . . . 164
7.2.5 Cas Euclidien . . . 165

7.3 Une interprétation géométrique . . . 166
7.3.1 Cas projectif . . . 166
7.3.2 Cas affine . . . 170
7.3.3 Cas métrique et Euclidien . . . 170
7.4 Extraction du mouvement à partir d'une matrice de mouvement pour droites 3D . . . 171
7.4.1 Cas projectif . . . 171
7.4.2 Cas affine . . . 172
7.4.3 Cas métrique . . . 173
7.4.4 Cas Euclidien . . . 173
7.5 La matrice de passage projectif-métrique pour droites 3D . . . 174
7.6 Conclusions . . . 174

8 Alignement de reconstructions de droites 177
8.1 Introduction . . . 178
8.2 Cas minimaux . . . 179
8.3 Notations, scénario et désignation des méthodes proposées . . . 180
8.3.1 Notations et scénario . . . 180
8.3.2 Désignation et résumé des méthodes proposées . . . 180
8.4 Maximum de vraisemblance . . . 180
8.4.1 Formulation algébrique . . . 180
8.4.2 Optimisation non-linéaire . . . 183
8.5 Méthodes linéaires et quasi-linéaires . . . 184
8.5.1 Cas projectif . . . 184
8.5.2 Cas affine . . . 193
8.5.3 Cas métrique et Euclidien . . . 197
8.6 Méthodes non-linéaires . . . 198
8.7 Résultats expérimentaux . . . 199
8.7.1 Données simulées . . . 200
8.7.2 Données réelles . . . 203
8.8 Conclusions et perspectives . . . 211

9 Conclusions, perspectives et problèmes ouverts 213
9.1 Conclusions . . . 214
9.2 Perspectives . . . 215
9.3 Problèmes ouverts . . . 216

A Détection de plans 217
A.1 Introduction . . . 218
A.2 Etat de l'art . . . 219
A.3 Segmentation en plans . . . 220
A.3.1 Description du processus . . . 220
A.3.2 Segmentation par échantillonage aléatoire . . . 220
A.4 Résultats expérimentaux . . . 223
A.4.1 Données simulées . . . 225
A.4.2 Données réelles . . . 226
A.5 Conclusions et perspectives . . . 232

B Géométrie et reconstruction dynamiques – Mouvements linéaires, coplanaires et convergents 235
B.1 Introduction . . . 236
B.2 La géométrie dynamique et son estimation . . . 238
B.2.1 Le cas de deux vues : le tenseur-C7 . . . 238
B.2.2 Le cas multi-vues : le tenseur-C5 . . . 240

B.3 Estimation consistante des géométries dynamiques et statiques 242
B.3.1 Le cas de deux vues 243
B.3.2 Le cas multi-vues 244
B.4 Résultats expérimentaux 246
B.5 Conclusions et perspectives 247

C Correction de Plücker **249**
C.1 Introduction 249
C.2 Formulation du problème 250
C.3 Résolution 250

D Autres travaux **253**

Bibliographie **255**

Table des figures

2.1 Le modèle de caméra perspectif. 24
2.2 La géométrie épipolaire. 27
2.3 Interprétation géométrique des coordonnées de Plücker. 33

3.1 Reconstruction d'un point par intersection de rayons de projection. 36
3.2 Reconstruction de point par ajustement de faisceaux. 40
3.3 Reconstruction de droite par ajustement de faisceaux. 41
3.4 Rôle de la fonction de paramétrisation dans l'optimisation non-linéaire. 45
3.5 Structure des matrices Jacobienne et Hessienne (approximation de Gauss-Newton) lors de l'ajustement de faisceaux. 49

4.1 Rôle des homographies de plan dans la factorisation partielle de la matrice fondamentale. . . . 54
4.2 Rôle des transformations épipolaires étendues dans la factorisation totale de la matrice fondamentale. 56
4.3 Représentation orthonormale de la matrice fondamentale et optimisation non-linéaire. 64
4.4 Erreur de reprojection et temps de calcul pour différentes distances scène à caméras. 74
4.5 Erreur de reprojection et temps de calcul pour différentes distances focales des caméras. . . . 74
4.6 Erreur de reprojection et temps de calcul pour différents niveaux de bruit image. 75
4.7 Erreur de reprojection et temps de calcul pour différents nombres de points. 76
4.8 Images utilisées pour comparer différentes méthodes d'estimation de la matrice fondamentale. 77

5.1 Séquence "game system I". 80
5.2 Modélisation plane par morceaux de la scène "game system I". 83
5.3 Les différentes étapes de l'initialisation d'une structure plane par morceaux. 86
5.4 Paramétrisation minimale d'un point sur un plan dans le cas de deux vues. 92
5.5 Paramétrisation minimale d'un point sur deux plans dans le cas de deux vues. 93
5.6 Illustration du fait qu'un point sur trois plans est en général déterminé de manière unique. . . . 95
5.7 Structure des matrices Jacobienne et Hessienne (approximation de Gauss-Newton) lors de l'ajustement de faisceaux standard non contraint. 96
5.8 Structure des matrices Jacobienne et Hessienne (approximation de Gauss-Newton) lors de l'ajustement de faisceaux plan par morceaux. 97
5.9 Structure des matrices Hessiennes (approximation de Gauss-Newton) lors de l'ajustement de faisceaux standard non contraint et plan par morceaux. 98
5.10 Erreur basée image minimisée par la méthode linéaire de reconstruction multi-vues de plans. . 99
5.11 Erreur 3D pour différents niveaux de bruit image et écartements entre les caméras. 107
5.12 Erreur 3D pour différents nombres de points et nombres de vues. 108
5.13 Erreur 3D pour différentes distances scène à caméras et taux de perturbation de la planarité. . . 108
5.14 Erreur 3D sur la position des caméras pour différents niveaux de bruit image et nombres de vues. 110
5.15 Système global de reconstruction d'une scène plane par morceaux. 111
5.16 Séquence de la maison. 112
5.17 Vues du modèle 3D métrique pour la séquence de la maison. 113
5.18 Reprojection du modèle 3D métrique sur les images originales de la maison. 114

5.19 Vues du modèle 3D métrique texturé pour la séquence de la maison. 114
5.20 Vues du modèle 3D projectif texturé pour la séquence “game system I”. 115
5.21 Vues du modèle 3D métrique texturé pour la séquence “game system I”. 116
5.22 Mesures de qualité métriques sur le modèle 3D pour la séquence “game system I”. 117
5.23 Séquence des cubes. 117
5.24 Points image et plans modélisés pour la séquence des cubes. 117
5.25 Vues du modèle 3D projectif texturé pour la séquence des cubes. 118
5.26 Vues du modèle 3D métrique texturé pour la séquence des cubes. 119
5.27 Mesures de qualité métriques sur le modèle 3D pour la séquence des cubes. 120

6.1 Représentation orthonormale des droites 3D et optimisation non-linéaire. 134
6.2 Interprétation géométrique de la représentation orthonormale. 139
6.3 Interprétation géométrique des paramètres de mise à jour de la représentation orthonormale. . 140
6.4 Erreur de reprojection pour différents niveaux de bruit image et nombres de droites. 142
6.5 Erreur de reprojection pour différents nombres d’images et distances scènes à caméras. 143
6.6 Séquence des livres I. 144
6.7 Vues du modèle 3D métrique pour la séquence des livres I. 145
6.8 Reprojection des droites dans les images de la séquence des livres I. 146
6.9 Séquence des livres II. 147
6.10 Vues du modèle 3D métrique pour la séquence des livres II. 148
6.11 Séquence des boîtes I. 151
6.12 Vues du modèle 3D métrique pour la séquence des boîtes I. 152
6.13 Séquence des boîtes II. 153
6.14 Vues du modèle 3D métrique pour la séquence des boîtes II. 154

7.1 Scénario pour l’interprétation géométrique des matrices de mouvement pour droites 3D. . . . 167
7.2 Scénario pour l’interprétation géométrique de la matrice d’homographie pour droites 3D. . . . 168

8.1 Scénario considéré pour l’alignement de reconstructions de droites. 181
8.2 Principe de la formulation de contraintes 3D linéaires sur les coefficients de la matrice de mouvement usuelle à partir de correspondances de droites. 185
8.3 Principe de la formulation de contraintes basées image linéaires sur les coefficients de la matrice de mouvement usuelle à partir de correspondances de droites. 187
8.4 Principe de la formulation de contraintes 3D linéaires sur les coefficients de la matrice de mouvement pour droites 3D à partir de correspondances de droites. 189
8.5 Principe de la formulation de contraintes basées image linéaires sur les coefficients de la matrice de mouvement pour droites 3D à partir de correspondances de droites. 191
8.6 Erreur de reprojection pour différents niveaux de bruit image. 201
8.7 Erreur de reprojection pour différents nombres de droites. 202
8.8 Séquence des livres I : une image et une vue du modèle 3D métrique. 204
8.9 Séquence des livres II : une image et une vue du modèle 3D métrique. 204
8.10 Alignement des séquences des livres I et des livres II – 1. 205
8.11 Alignement des séquences des livres I et des livres II – 2. 205
8.12 Séquence des boîtes I : une image et une vue du modèle 3D métrique. 206
8.13 Séquence des boîtes II : une image et une vue du modèle 3D métrique. 206
8.14 Alignement des séquences des boîtes I et des boîtes II. 207
8.15 Séquence de la pièce de bateau. 208
8.16 Reprojection après transfert sur les images originales de la pièce de bateau. 209
8.17 Vues du modèle 3D métrique pour la séquence de la pièce de bateau. 210

A.1 Exemple de scène plane par morceaux, la séquence “game system II” : une image et une vue du modèle 3D métrique. 218

A.2 Différence entre la segmentation en ensembles disjoints et en ensembles non disjoints. 221
A.3 Différence entre la segmentation avec un critère géométrique individuel et un critère photométrique dense. 223
A.4 Evaluation de la pertinence d'un plan hypothétique selon un critère photométrique dense. . . . 224
A.5 Erreur dans le nombre de plans détectés pour différents niveaux de bruit image, nombres de points et taux de perturbation de la planarité. 225
A.6 Séquence du modem. 226
A.7 Points d'intérêt et vues du modèle 3D métrique pour la séquence du modem. 227
A.8 Segmentation automatique en plans pour la séquence du modem. 228
A.9 Vues du modèle 3D métrique pour la séquence du modem. 229
A.10 Vues du modèle 3D métrique texturé pour la séquence du modem. 229
A.11 Séquence "game system II". 230
A.12 Vues du modèle 3D métrique pour la séquence "game system II". 230
A.13 Segmentation automatique en plans pour différents rayons de dispersion, pour la séquence "game system II". 231
A.14 Vues du modèle 3D métrique texturé pour la séquence "game system II". 232

B.1 Séquence de la route. 236
B.2 Le scénario dynamique considéré. 237
B.3 Séquence des jouets. 238
B.4 Droites de mouvement estimées indépendamment par paires d'images pour la séquence des jouets. 240
B.5 Droites de mouvement estimées par paires d'images de manière consistante pour la séquence des jouets. 242
B.6 Transfert des points dynamiques pour la séquence des jouets. 245
B.7 Transfert des points dynamiques et statiques pour la séquence de la route. 246

Liste des tableaux

2.1 Caractérisation des transformations 3D dans différents espaces. 10
2.2 Algorithme d'estimation quasi-linéaire d'une homographie 2D. 18

4.1 Résumé de la représentation orthonormale de la matrice fondamentale. 69
4.2 Implantation d'un algorithme d'ajustement de faisceaux basé sur la représentation orthonormale de la matrice fondamentale. 71
4.3 Erreur de reprojection et temps de calcul pour des paires d'images de la figure 4.8. 76

5.1 Reconstructabilité des points selon le nombre de vues et le nombre de plans. 84
5.2 Résumé de l'état de l'art de méthodes d'ajustement de faisceaux plan par morceaux. 87
5.3 Paramétrisation minimale d'un point sur un plan dans le cas multi-vues. 104
5.4 Détermination expérimentale du seuil de rupture. 109
5.5 Erreur de reprojection et nombre d'itérations pour la séquence de la maison. 113
5.6 Mesures de qualité métriques pour la séquence de la maison. 115

6.1 Différentes représentations des droites 3D. 126
6.2 Matrices de projection pour coordonnées de Plücker. 129
6.3 Algorithme de triangulation linéaire LIN. 130
6.4 Algorithme de triangulation quasi-linéaire QLIN1. 131
6.5 Algorithme de triangulation quasi-linéaire QLIN2. 132
6.6 La représentation orthonormale des droites 3D. 137
6.7 Exemples d'estimations contraintes d'une droite 3D à l'aide de la représentation orthonormale. 138
6.8 Erreur de reprojection pour la séquence des livres I. 145
6.9 Erreur de reprojection pour la séquence des livres II. 148
6.10 Erreur de reprojection pour la séquence des boîtes I. 149
6.11 Erreur de reprojection pour la séquence des boîtes II. 149

7.1 Extraction d'une matrice d'homographie à partir d'une matrice d'homographie pour droites 3D. 171
7.2 Extraction d'une matrice d'affinité à partir d'une matrice d'affinité pour droites 3D. 173
7.3 Extraction d'une matrice de similitude à partir d'une matrice de similitude pour droites 3D. . . 173
7.4 Extraction d'une matrice de mouvement rigide à partir d'une matrice de mouvement rigide pour droites 3D. 174

8.1 Cas minimaux pour l'estimation d'un mouvement 3D à partir de correspondances de droites 3D. 179
8.2 Méthodes proposées pour l'alignement de reconstructions de droites. 182

A.1 Modélisation des plans à l'issue de l'étape de segmentation. 220
A.2 Erreur de reprojection pour la séquence du modem. 227
A.3 Mesures de qualité métriques pour la séquence "game system II". 232

C.1 Algorithme pour la correction de Plücker. 252

CHAPITRE

1

INTRODUCTION

La *vision par ordinateur* a pour vocation d'inférer de l'information à partir d'images, en général des photographies ou des films vidéos. Les applications sont multiples : la vidéo-surveillance, la création de modèles 3D numériques, la recherche d'information dans des bases d'images ou de films vidéos, la chirurgie assistée par ordinateur, etc. Les problèmes sont en général posés comme des problèmes inverses. Parallèlement aux avancées théoriques en vision par ordinateur, le développement des technologies qui en sont le produit a été rendu possible grâce à la démocratisation d'appareils photos et de caméras vidéo numériques intégrant des systèmes optiques de grande qualité. L'expansion des systèmes informatiques a fourni la puissance de calcul nécessaire au traitement des données.

La *vision 3D par ordinateur* est un sous-ensemble de la vision par ordinateur, dont le but est d'inférer de l'information tridimensionnelle à partir d'images. La photogrammétrie est un domaine proche, ayant un but similaire. Lorsque nous observons notre environnement, nous sommes capables, en général, d'estimer la distance qui nous sépare des différents objets, de comparer ces distances, d'estimer des vitesses. De manière générale, nous obtenons de l'information de nature tridimensionnelle. Le but de la vision 3D par ordinateur est de réaliser une tâche similaire, à l'aide d'un périphérique de capture d'image (par exemple un appareil photo ou un caméscope numérique), ou *caméra*, relié à un ordinateur. Plus précisément, à partir de deux images (ou plus) d'une même scène, prises de points de vue différents, le but est d'inférer des informations tridimensionnelles, telles des profondeurs, souvent assimilées à la reconstruction d'un *modèle numérique 3D* de la scène. Il est courant en vision 3D par ordinateur, de calculer les positions 3D des caméras conjointement à toute inférence d'information tridimensionnelle.

La photogrammétrie est un domaine plus ancien et donc plus mature que la vision 3D par ordinateur. De nombreuses applications en découlent, principalement la réalisation de mesures tridimensionnelles précises et fiables. La photogrammétrie diffère de la vision 3D par ordinateur dans le sens où l'environnement est souvent supposé contrôlable. Par exemple, on suppose couramment en photogrammétrie, que les positions 3D des caméras sont connues, au moins approximativement. Les photogrammètres utilisent souvent des modèles mathématiques poussés afin d'améliorer la précision des résultats, et cherchent à quantifier leur fiabilité.

Cette thèse s'inscrit dans le domaine de la vision 3D par ordinateur, avec quelques contributions à des domaines communs avec la photogrammétrie. Par exemple, nous ne supposons pas les positions 3D des caméras connues a priori.

Nous nous intéressons à la reconstruction de modèles 3D à partir d'images, ou tout simplement la *reconstruction*. Un modèle 3D reconstruit peut être utilisé pour générer de nouvelles vues de la scène, et peut être modifié à l'aide d'un outil de modélisation interactive. Des objets peuvent être modifiés, ajoutés ou supprimés. De même, les couleurs, les apparences et l'atmosphère de la scène peuvent être changés. Les applications sont nombreuses, en particulier, les jeux vidéos, la navigation autonome, l'architecture, la réalité augmentée, la post-production cinématographique, etc.

Nous nous intéressons plus particulièrement à la reconstruction *optimale* de modèles 3D. La définition du terme optimal requiert de poser plus précisément le cadre de l'approche, en particulier d'en définir les données d'entrée, et de caractériser plus précisément les *tâches de reconstruction*.

1.1 Modélisation et données d'entrée

Il nous apparaît important de définir en premier lieu la représentation du modèle 3D. Le choix de cette représentation est lié à plusieurs facteurs, notamment l'utilisation qui sera faite du modèle 3D. Avant d'aller plus avant, rappelons qu'un modèle 3D est souvent constitué de la scène observée, désignée par le terme *structure*, et des caméras ayant acquis les images, désignées par le terme *mouvement*.

La représentation de la structure peut être inspirée des techniques développées dans le domaine de la synthèse d'images. Un choix standard consiste à utiliser un ensemble de facettes planes, reposant sur des *primitives*, telles des points ou des droites. L'utilisation d'images de textures et la modélisation de la réflectance de ces facettes permet de conférer un aspect photoréaliste à la structure. Une autre représentation est la représentation volumique, basée sur une discrétisation de l'espace en volumes élémentaires.

Nous utilisons différentes représentations, constituées de primitives, et parfois de plans reposants sur ces primitives. Les projections des primitives sont directement mesurables dans les images : une primitive 3D se projette en général en une primitive de même nature dans les images.

Nous ne donnons pas ici de détails sur la représentation du mouvement. Nos contributions sont formulées dans la mesure du possible pour le cas général de la *reconstruction projective*, c'est à dire à partir de caméras non calibrées, et sont spécialisées, lorsque ceci n'est pas trivial, au cas de caméras calibrées.

Les données d'entrée que nous utilisons sont des correspondances de primitives image, c'est à dire issues de la projection d'une même primitive 3D. D'autres possibilités de données d'entrée sont des contours ou silhouettes d'objets, ou le flot optique, c'est à dire le déplacement relatif de chaque pixel entre les images. Plus précisément, nous utilisons des correspondances de points ou de droites et éventuellement des contraintes géométriques entre ces primitives, de type coplanarité. Le problème de la détection et de la mise en correspondance des primitives n'est pas abordé dans cette thèse : nos algorithmes se situent en aval de cette étape. La position des primitives image est affectée par un *bruit de détection*. Le problème de la détection automatique de relations géométriques, telle la coplanarité, est abordé comme une contribution annexe de cette thèse.

1.2 Stratégies et tâches de reconstruction

Cette section examine quelques stratégies de reconstruction à un niveau conceptuel global. La reconstruction est l'inférence d'un modèle 3D à partir d'images. Lorsqu'un petit nombre d'images est traité, le modèle 3D complet est souvent obtenu par *inférence directe* à partir des images. Ce que nous appelons inférence directe, est la brique de base de la reconstruction, consistant à passer des primitives image aux primitives 3D et éventuellement aux caméras. L'inférence directe d'un modèle 3D s'applique mal à un grand nombre d'images. La complexité des structures algébriques encapsulant la géométrie multi-vues s'accroît rapidement avec le nombre d'images. D'autres stratégies de reconstruction sont alors mises en œuvre. Elles reposent sur l'inférence d'un ou plusieurs modèles 3D partiels à partir de petites sous-séquences d'images. La première stratégie consiste à compléter un modèle 3D en inférant itérativement chaque caméra et de nouvelles primitives par rapport au modèle 3D existant. La deuxième stratégie consiste à *fusionner* plusieurs modèles 3D partiels. La fusion de modèles 3D requiert leur *alignement*, et sera désignée par ce terme.

Ces deux stratégies conduisent aux *tâches de reconstruction* suivantes :

▷ *inférence directe* d'un modèle 3D ;
▷ inférence d'une caméra et de primitives, par rapport à un modèle 3D partiel ;
▷ *alignement* de modèles 3D partiels.

Cette thèse aborde les problèmes de l'inférence directe d'un modèle 3D et de l'alignement de modèles 3D partiels.

1.3 Reconstruction optimale

On peut maintenant se poser les deux questions suivantes :
▷ quelle est la définition du terme *optimal* ?
▷ comment définir un *critère* ou *fonction de coût*, dont la minimisation conduirait à un résultat optimal ?

Nous utilisons le terme optimal au sens statistique du *maximum de vraisemblance*. C'est une définition standard en vision par ordinateur, basée entre autres sur l'hypothèse que toutes les configurations géométriques sont équiprobables. Le choix du maximum de vraisemblance est raisonnable sous l'hypothèse que le bruit sur les données est faible (par exemple de l'ordre de quelques pixels sur la position de points image), ce qui est souvent le cas en vision par ordinateur, et en particulier pour les problèmes que nous étudions. En faisant l'hypothèse que la distribution du bruit de mesure est centrée, Gaussienne, identique et indépendante, l'estimation au maximum de vraisemblance est donnée par la minimisation de l'*erreur de reprojection*. Cette dernière décrit comment le modèle 3D explique les mesures.

Nous ne remettons pas ces hypothèses standards en question, et assimilons l'estimation optimale à la minimisation de l'erreur de reprojection. C'est une fonction de coût couramment utilisée en vision 3D par ordinateur et en photogrammétrie.

Le problème du calcul d'une solution est une instance d'une classe de problèmes plus générale, qualifiée de manière différente selon les communautés, entre autres, erreurs-dans-les-variables[1], moindres carrés totaux[2], ou régression en distances orthogonales[3].

L'erreur de reprojection s'exprime comme une somme de carrés non-linéaires en termes des paramètres de la structure et du mouvement. L'estimation optimale consiste à minimiser l'erreur de reprojection sur tous ces paramètres. Il n'est pas possible de calculer une solution de manière directe : la reconstruction est effectuée de manière itérative, par mises à jour successives des paramètres, ayant pour but de minimiser l'erreur de reprojection. Nous utilisons des techniques d'optimisation locale, permettant d'améliorer une solution initiale sous-optimale (c'est à dire ne minimisant pas l'erreur de reprojection). Les algorithmes ainsi obtenus sont rassemblés sous le vocable *ajustement de faisceaux*[4]. Notons que l'erreur de reprojection ne dépend pas du niveau de calibrage interne (par exemple de la connaissance de la distance focale ou du point principal) des caméras.

Deux problèmes principaux sont soulevés :
▷ l'estimation d'une *solution initiale*, un point de départ pour l'optimisation non-linéaire itérative ;
▷ la conception d'une *paramétrisation algébrique* adaptée aux algorithmes d'optimisation non-linéaire.

Un ajustement de faisceaux est souvent utilisé comme étape finale d'un processus de reconstruction. Les étapes antérieures sont considérées comme l'initialisation. Notons que selon la stratégie de reconstruction, les étapes relevant de l'initialisation sont elles-mêmes des tâches de reconstruction, et font l'objet du calcul d'une solution initiale et éventuellement d'un ajustement de faisceaux.

Les problèmes de l'initialisation et de la paramétrisation sont détaillés ci-dessous.

1.3.1 Initialisation

Il n'existe pas de solution directe et optimale au problème de la reconstruction. Cependant, l'estimation itérative non-linéaire nécessite un point de départ, le plus proche possible de la solution optimale. L'initialisation

[1] "errors-in-variables" ou EIV en Anglais.
[2] "total least squares" ou TLS en Anglais.
[3] "orthogonal distance regression" ou ODR en Anglais.
[4] "bundle adjustment" en Anglais.

est donc une étape difficile, car l'erreur de reprojection est non-linéaire, et une solution initiale trop mauvaise provoque la convergence de l'ajustement de faisceaux vers un minimum local correspondant à une solution sous-optimale. L'approche classique est de formuler l'initialisation comme un problème de moindres carrés linéaires, permettant l'estimation directe d'une solution initiale. Plusieurs formulations sous bien sûr possibles, et conduisent à des résultats différents. L'analyse de la fonction de coût minimisée par un algorithme linéaire permet en général de comprendre ses performances. A ce stage, il semble naturel de se poser la question suivante :

▷ existe-t-il un critère et une méthodologie pour le choix d'une fonction de coût linéaire ?

Notons que la comparaison expérimentale permet de départager les différentes fonctions de coût.

1.3.2 Paramétrisation

La phase d'ajustement de faisceaux fait appel à des algorithmes d'optimisation de fonctions de coût de moindres carrés non-linéaires, souvent dérivés de l'algorithme de Newton. L'application de ces algorithmes au problème de l'ajustement de faisceaux requiert une paramétrisation algébrique du problème, c'est à dire un jeu de paramètres sur lesquels agir, afin de faire évoluer la fonction de coût. Les représentations naturelles des entités 3D, caméras, points, droites, plans, etc., ne constituent pas toujours une paramétrisation adaptée. Ceci soulève la question suivante :

▷ quelle est la définition d'une paramétrisation adaptée à l'ajustement de faisceaux ?

1.4 Contributions

Cette thèse regroupe une grande partie de nos contributions aux domaines de la vision 3D par ordinateur et de la photogrammétrie. Ces contributions concernent la résolution de certaines tâches précises de reconstruction. Des problèmes variés sont abordés, et les contributions sont, de ce point de vue, indépendantes. Par contre, elles sont étroitement liées par des concepts plus généraux que les tâches de reconstruction, formant les lignes directrices, en d'autres termes le fil conducteur de cette thèse. Nous présentons ces lignes directrices, puis donnons le détail des contributions plus précises.

1.4.1 Lignes directrices

L'élaboration de nos contributions a été guidée par les deux concepts clefs suivants. Ces concepts sont issus du questionnement précédent concernant la méthodologie de la définition d'une fonction de coût linéaire, et la définition de critères de qualité associés à une paramétrisation :

▷ **Formuler des méthodes d'initialisation minimisant des fonctions de coût les plus proches possible de l'erreur de reprojection.** Comme discuté ci-dessus, la solution initiale conditionne souvent la convergence d'un ajustement de faisceaux vers le minimum de l'erreur de reprojection recherché. Intuitivement, plus la forme de la fonction de coût est proche de celle de l'erreur de reprojection, meilleure sera l'initialisation. Cette définition est floue, mais peut être concrétisée par des critères plus précis, par exemple une fonction de coût basée sur une distance Euclidienne pour comparer les mesures et les prédictions est a priori plus souhaitable qu'une autre basée sur une distance algébrique. Les résultats expérimentaux obtenus confirment cette intuition. De manière générale, nous tentons de minimiser directement l'erreur de reprojection quand cela est possible, ou une version linéarisée, préservant sa forme. Nous avons apporté un soin particulier pour que nos méthodes de reconstruction de plans, de triangulation de droites ou encore d'alignement de reconstruction de droites, minimisent des fonctions de coût répondant à ces critères ;

▷ **Concevoir des paramétrisations adaptées aux algorithmes d'ajustement de faisceaux.** La phase d'ajustement de faisceaux nécessite une paramétrisation du problème. Cette paramétrisation influence entre autres le temps de calcul, la stabilité numétrique de l'estimation et la facilité d'implantion de l'algorithme. Nous désirons que nos paramétrisations répondent à certains critères de qualité. Pour ce faire, une étude préliminaire à nos contributions a pour but de définir de tels critères. Ces derniers sont en-

suite exploités pour la conception de différentes paramétrisations. Un des critères que nous considérons comme important est la *minimalité* : est-ce que la paramétrisation permet de conduire l'optimisation sur un nombre de paramètres correspondant au nombre de degrés de liberté physiques du problème ? Si oui, on parlera alors d'*optimisation minimale*. Les autres critères de qualité que nous considérons sont, brièvement, la *linéarité* (la paramétrisation doit être le plus linéaire possible), l'*unicité* et les *contraintes internes* (le jeu de paramètres doit être de préférence unique, sans être soumis à des contraintes internes), la *complétude* (la paramétrisation doit pouvoir représenter toutes les instances de l'entité représentée), et d'autres que nous ne détaillons pas ici. Nous proposons des paramétrisations de la matrice fondamentale, des structures planes par morceaux, ou encore des droites 3D, tentant de satisfaire au mieux ces critères.

1.4.2 Paramétrisations et algorithmes

Nous listons ci-dessous les différentes tâches de reconstruction abordées dans cette thèse, et de manière plus pragmatique que ci-dessus, les paramétrisations et algorithmes proposés pour leur résolution :

- ▷ **Reconstruction de points et de deux caméras non calibrées.** Cette tâche de reconstruction est essentielle en vision 3D par ordinateur. Elle consiste à estimer la matrice fondamentale entre deux vues ainsi qu'un nuage de points dans l'espace projectif. Nous abordons le problème de l'ajustement de faisceaux, et de manière plus générale, de l'optimisation non-linéaire de la matrice fondamentale. Les paramétrisations existantes permettant l'optimisation minimale conduisent à des solutions algorithmiques compliquées ou sous-optimales. Nous proposons la **représentation orthonormale de la matrice fondamentale**, permettant l'estimation non-linéaire minimale de manière simple. Cette contribution est précédée de l'étude de ce que nous appelons la **factorisation totale de la matrice fondamentale**. Ce formalisme, basé sur les **transformations épipolaires étendues** que nous introduisons, permet de former la plupart des paramétrisations existantes ;
- ▷ **Reconstruction de plans, de points et de caméras.** Cette tâche de reconstruction consiste à intégrer à l'inférence directe de points et de caméras, des contraintes géométriques de coplanarité. Les méthodes existantes sont sous-optimales ou font appel à des techniques d'optimisation non-linéaire compliquées. Le problème de l'initialisation est rarement abordé. Basé sur une modélisation soigneuse des contraintes, nous donnons des méthodes pour la reconstruction initiale basées sur des fonctions de coût les plus proches possibles de l'erreur de reprojection. Notre algorithme est basé sur la reconstruction successive des caméras, des plans, puis des points. C'est cet ordre de reconstruction qui permet de concevoir des méthodes efficaces. En particulier, les points sont reconstruit directement sur les plans, en minimisant l'erreur de reprojection, par nos méthodes de **triangulation contrainte par coplanarité**. Nous proposons une paramétrisation minimale d'une structure plane par morceaux, permettant l'utilisation d'un algorithme d'optimisation non-linéaire standard (non contraint) pour l'ajustement de faisceaux. Nos méthodes nous permettent de mener une étude expérimentale, visant à déterminer les limites de validité des contraintes de coplanarité (il n'existe pas de surface réelle parfaitement plane) ;
- ▷ **Reconstruction de droites et de caméras.** La reconstruction de droites et de caméras est un problème classique en vision 3D par ordinateur et photogrammétrie. Les algorithmes d'initialisation existants ne minimisent pas l'erreur de reprojection. Nous proposons un algorithme de **triangulation**, minimisant l'erreur de reprojection. Cet algorithme est obtenu après une étude des différentes représentations des droites 3D. Cette étude met par ailleurs en lumière le fait que les représentations standards des droites 3D ne sont pas adaptées aux algorithmes d'ajustement de faisceaux. Nous proposons la **représentation orthonormale des droites 3D**, permettant comme dans le cas de la matrice fondamentale, l'estimation non-linéaire minimale de manière simple ;
- ▷ **Alignement de reconstructions de droites.** Nos contributions à cette tâche de reconstruction sont séparées en deux parties. La première partie est une contribution théorique à la géométrie algébrique des droites 3D. Plus précisément, nous étudions l'application d'un mouvement 3D à une droite 3D exprimée en coordonnées de Plücker. Nous proposons les **matrices de mouvement pour droites 3D**. Ce sont des matrices de taille (6×6), permettant d'appliquer un mouvement 3D directement aux coordonnées de Plücker d'une droite 3D. Notre formalisme permet de formuler les matrices de mouvement

pour droites 3D correspondant à différents types de mouvement : les matrices d'homographie, d'affinité, de similitude et de mouvement rigide pour droites 3D. La deuxième partie est centrée sur la conception d'algorithmes d'estimation d'un mouvement 3D à partir de correspondances de droites 3D. Après avoir défini la fonction de coût dont la minimisation conduit à un résultat optimal, nous proposons de nombreux algorithmes d'estimation, tentant de concilier simplicité et de ne pas introduire un biais trop fort par rapport à la fonction de coût précédemment définie. Ces algorithmes sont basés en partie sur le formalisme des matrices de mouvement pour droites 3D.

Organisation de la thèse. Nous avons donné ci-dessus une introduction générale aux problèmes étudiés dans cette thèse. Le chapitre 3 donne une vue plus précise sur les méthodes d'initialisation, les techniques d'ajustement de faisceaux, et les problèmes qui y sont associés. En particulier, nous y formulons l'erreur de reprojection pour la reconstruction de caméras, de points et de droites, et y formalisons le concept de paramétrisation, les critères de qualité d'une paramétrisation et le lien avec les algorithmes d'optimisation non-linéaire employés pour l'ajustement de faisceaux.

Le chapitre 2 présente les concepts de base et les notations utilisées au cours de la thèse. Il sert de référence, et nous y ferons de nombreux renvois. Il contient des notions standards, mais aussi des concepts parfois méconnus, tels les propriétés de la matrice des cofacteurs, le principe de l'optimisation quasi-linéaire ou encore un schéma d'estimation non-linéaire des rotations.

Chacun des chapitres suivants (excepté le chapitre de conclusions et perspectives) contient une introduction et un état de l'art détaillé sur le problème qui y est abordé. La plupart contiennent des résultats expérimentaux, obtenus sur des données simulées et réelles, ainsi que des perspectives de travail.

La répartition de nos contributions sur les différents chapitres est proche de l'ordre suivi dans la section précédente.

Le chapitre 4 considère le problème de la reconstruction de points et de deux caméras non calibrées. Le chapitre 5 aborde le problème de la reconstruction de points, de plans et de caméras. Le chapitre 6 propose des méthodes pour la reconstruction de droites et de caméras. Le chapitre 7 contient notre contribution théorique à la géométrie algébrique des droites 3D, axée sur la mise en évidence et l'étude des matrices de mouvement pour droites 3D. Le chapitre 8 étudie l'alignement de reconstructions de droites et de caméras. Le chapitre 9 donne nos conclusions, synthétise des perspectives de travail sur les sujets abordés, et examine quelques problèmes ouverts.

Enfin, quatre annexes terminent le manuscrit. L'annexe A est liée à la reconstruction de plans, de points et de caméras. Elle aborde le problème de la détection automatique de plans. L'annexe B considère l'étude de la géométrie et de la reconstruction d'une scène partiellement dynamique et partiellement statique. L'annexe C est liée à la reconstruction de droites et de caméras. Nous y développons une méthode auxiliaire nécessaire à nos algorithmes de triangulation de droites. Finalement, l'annexe D résume d'autres travaux, non détaillés au cours du manuscrit.

CHAPITRE

2

ELÉMENTS DE BASE

Ce chapitre expose les éléments de base nécessaires à la lecture de cette thèse. Nous y définissons les conventions et notations utilisées et développons l'utilisation de certains opérateurs et décompositions de matrices. Nous décrivons des mesures de distance Euclidiennes et algébriques et le principe des algorithmes quasi-linéaires. Nous examinons les propriétés des matrices de rotation et un schéma d'estimation. Nous présentons le modèle de caméra utilisé, ainsi que la géométrie épipolaire, les homographies de plan et différents types de mouvements 3D. Finalement, nous décrivons les coordonnées de Plücker des droites 3D. Pour compléter la lecture de ce chapitre, nous renvoyons aux livres de Faugeras [58], Faugeras et Luong [60] et Hartley et Zisserman [93].

2.1 Conventions et notations

Dans cette thèse, des entités géométriques telles des points, droites, plans sont considérées en différentes dimensions. Ces entités sont désignées par des caractères gras et italiques. Par exemple, les points sont souvent notés $\boldsymbol{Q}$ et les plans $\boldsymbol{\Pi}$. Sauf mention contraire explicite, toutes ces entités géométriques sont exprimées algébriquement par des vecteurs de coordonnées homogènes, c'est à dire définis à un facteur multiplicatif non nul près. L'égalité à un facteur près est notée $\sim$. La notation employée distingue les entités géométriques de leurs coordonnées exprimées dans une certaine base. Les vecteurs de coordonnées sont souvent notés avec la même lettre que l'entité géométrique, écrit avec un caractère gras et droit. Par exemple, les coordonnées du point $\boldsymbol{Q}$ dans une certaine base sont notées $\mathbf{Q}$, et celles du plan $\boldsymbol{\Pi}$ sont notées $\mathbf{\Pi}$. Nous ne faisons pas de différence formelle entre une matrice représentant une transformation géométrique et la transformation elle-même. Par exemple, H désigne une homographie *et* la matrice représentant cette homographie.

Les coefficients d'un vecteur $\mathbf{Q}$ sont notés avec le même caractère, en italique, et un indice indiquant sa position : $\mathbf{Q}^\mathsf{T} = (Q_1 \;\; Q_2 \;\; \dots)$, où $^\mathsf{T}$ désigne la transposition d'un vecteur ou d'une matrice. Les vecteurs de coordonnées sont souvent partitionnés pour distinguer leur dernier coefficient : $\mathbf{Q}^\mathsf{T} \sim (\bar{\mathbf{Q}}^\mathsf{T} \;\; Q)$.

La taille des entités algébriques est notée à l'aide de parenthèses, par exemple la taille du vecteur-4 $\mathbf{Q}$ est notée (4×1). Les indices désignent soit la taille d'un vecteur (par exemple $\mathbf{Q}_{(4\times1)}$), soit un élément particulier parmi un ensemble (par exemple $\mathbf{Q}_j$ sont les coordonnées du $j^{\text{ème}}$ point). L'indice i est utilisé pour les caméras, j pour les points, j ou k pour les droites et l pour les plans. Le nombre total d'images est noté n, de points m, de droites m ou p et de plans z. Lorsque l'on veut désigner les coefficients d'un élément particulier d'un ensemble, on sépare les indices par des virgules, par exemple le $3^{\text{ème}}$ coefficient des coordonnées du $j^{\text{ème}}$ point est noté $\mathbf{Q}_{j,3}$.

Les entités 3D sont souvent notées avec des caractères majuscules et leurs observations dans des images avec la même lettre en minuscule. Ainsi, si $\mathbf{Q}_j$ représente les coordonnées de $\boldsymbol{Q}_j$, le $j^{\text{ème}}$ point, le point correspondant observé dans la $i^{\text{ème}}$ image est noté $\boldsymbol{q}_{ij}$ et ses coordonnées $\mathbf{q}_{ij}$.

Les matrices sont désignées par des caractères sans sérif, par exemple A. L'opération de vectorisation par ligne d'une matrice est notée vect. Par exemple, si les lignes de la matrice $\mathsf{A}_{(r\times c)}$ sont notées $\mathbf{r}_t$:

$$\mathsf{A}_{(r\times c)} \;=\; \begin{pmatrix} \mathbf{r}_1^\mathsf{T} \\ \cdots \\ \mathbf{r}_r^\mathsf{T} \end{pmatrix}, \quad \text{vect}(\mathsf{A})^\mathsf{T} \;=\; (\mathbf{r}_1^\mathsf{T} \;\; \dots \;\; \mathbf{r}_r^\mathsf{T}).$$

Le symbole $^{-\mathsf{T}}$ désigne la transposée de l'inverse (ou de manière équivalente, l'inverse de la transposée) d'une matrice, * la matrice des cofacteurs (ses propriétés sont étudiées en §2.3) et $^{\mathsf{T}\star}$ la transposée de la matrice des cofacteurs (ou de manière équivalente, la matrice des cofacteurs de la transposée). Le produit vectoriel est noté $\wedge$. La matrice $[\mathbf{x}]_\wedge$ est définie telle que pour tout vecteur-3 $\mathbf{y}$, $[\mathbf{x}]_\wedge \mathbf{y} = \mathbf{x} \wedge \mathbf{y}$:

$$[\mathbf{x}]_\wedge \;=\; \begin{pmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{pmatrix}.$$

Le produit mixte de trois vecteurs $\mathbf{a}$, $\mathbf{b}$ et $\mathbf{c}$ est noté $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ et défini par $(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \mathbf{a}^\mathsf{T}(\mathbf{b} \wedge \mathbf{c})$. Il est égal au déterminant de la matrice $(\mathbf{a} \;\; \mathbf{b} \;\; \mathbf{c})$. Les vecteurs et matrices nuls sont notés respectivement $\mathbf{0}$ et 0. La matrice identité est notée I. Leur taille est parfois indiquée lorsque le contexte ne la donne pas.

L'espace projectif de dimension p est noté $\mathbb{P}^p$. Les groupes de Lie des matrices orthonormales et de rotation de dimension p sont respectivement notés $O(p)$ et $SO(p)$. La norme $\mathcal{L}_2$ d'un vecteur $\mathbf{Q}$ est notée $\|\mathbf{Q}\|$. Elle est donnée par $\|\mathbf{Q}\|^2 = \mathbf{Q}^\mathsf{T}\mathbf{Q}$. Cette notation désigne de même la norme de Frobenius d'une matrice A, définie par $\|\mathsf{A}\|^2 = \|\text{vect}(\mathsf{A})\|^2 = \text{vect}(\mathsf{A})^\mathsf{T}\text{vect}(\mathsf{A})$.

Lors de la reconstruction à partir d'images, les entités prédites par le modèle reconstruit sont notées tel $\hat{\boldsymbol{x}}$. Par exemple, $\hat{\boldsymbol{q}}_{ij}$ désigne la prédiction du $j^{\text{ème}}$ point dans la $i^{\text{ème}}$ image. Certaines notations sont génériques dans le sens où elles sont utilisées pour plusieurs problèmes et n'ont qu'un sens "local". Par exemple, certains systèmes linéaires sont notés $\mathsf{A}\mathbf{x} = \mathbf{b}$ et certaines fonctions de coût $\mathcal{C}$.

2.2 Espaces Euclidien, métrique, affine et projectif

Dans cette thèse, nous travaillons avec plusieurs espaces géométriques : les espaces Euclidien, métrique, affine et projectif. Nous caractérisons brièvement ces espaces et les changements de base associés. Le raisonnement est en 3D mais reste valable dans d'autres dimensions, en particulier en 2D. La généralisation se fait en remplaçant les plans par des hyperplans.

L'espace Euclidien est celui qui nous entoure : les transformations sont rigides et les grandeurs physiques exprimées dans une unité de mesure. L'espace métrique est l'espace Euclidien où l'échelle globale a été changée[1]. En particulier, il n'est possible d'obtenir une reconstruction Euclidienne que lorsque la mesure d'une longueur réelle est introduite. En l'absence d'une telle mesure, seule une reconstruction métrique peut être obtenue à partir de caméras. En espace affine, les rapports de longueurs et les angles n'ont pas de sens, mais le parallélisme en a un. En espace projectif, toutes ces caractéristiques sont perdues. Le tableau 2.1 résume les invariants de ces différents espaces de manière plus complète.

Les coordonnées homogènes permettent de modéliser les points à l'infini. Topologiquement et algébriquement, l'infini est modélisé par un plan appelé le *plan à l'infini*, noté $\boldsymbol{\Pi}_\infty$, et dont les coordonnées sont :

$$\boldsymbol{\Pi}_\infty^\mathsf{T} \sim (0\ 0\ 0\ 1).$$

Un point $\boldsymbol{Q}$ de coordonnées $\mathbf{Q}^\mathsf{T} \sim (\ \bar{\mathbf{Q}}^\mathsf{T}\quad Q)$ se situe à l'infini si et seulement si $\boldsymbol{\Pi}_\infty^\mathsf{T}\mathbf{Q} = 0$, ce qui est équivalent à $Q = 0$. La quatrième coordonnée d'un point indique donc son appartenance au plan à l'infini. Une autre propriété est que les coordonnées $\boldsymbol{\Pi}^\mathsf{T} \sim (\ \bar{\boldsymbol{\Pi}}^\mathsf{T}\quad \Pi)$ de tout plan $\boldsymbol{\Pi}$ différent du plan à l'infini satisfont $\|\bar{\boldsymbol{\Pi}}\| \neq 0$. En espaces Euclidien, métrique et affine, le plan à l'infini correspond au "vrai" plan à l'infini. Ceci n'est pas toujours vrai en espace projectif, car les transformations projectives ne laissent pas le plan à l'infini invariant. Le tableau 2.1 résume les caractéristiques des transformations, c'est à dire les changements de base, dans ces différents espaces. Ces transformations sont modélisées par des matrices (4×4), permettant d'appliquer le mouvement directement aux coordonnées homogènes des points. Si $\mathbf{Q}$ représente les coordonnées homogènes d'un point dans une certaine base de l'espace, $\mathsf{G}\mathbf{Q}$ représente les coordonnées du même point dans une autre base de l'espace. Les transformations Euclidiennes sont des mouvements rigides, constitués d'une rotation et d'une translation. Les transformations métriques sont des similitudes, obtenues en ajoutant un facteur d'échelle aux transformations Euclidiennes. Les transformations affines généralisent le bloque (3×4) supérieur de la matrice et les transformations projectives généralisent toute la matrice. Pour illustrer certains raisonnements valables sur plusieurs types de mouvement, nous utilisons une *matrice de mouvement générique* (4×4) notée G.

2.3 La matrice des cofacteurs

La *matrice des cofacteurs* d'une matrice carrée S est notée $\mathsf{S}^\star$. Nous en examinons certaines propriétés, dont les preuves découlent de sa définition. La matrice des cofacteurs est surtout utilisée lors de la manipulation de droites 3D, aux chapitres 6, 7 et 8. Cette opération est définie quel que soit le rang de la matrice S. Lorsque ce rang est plein, elle est définie par :

$$\mathsf{S}^\star = \det(\mathsf{S})\mathsf{S}^{-\mathsf{T}}, \tag{2.1}$$

ce qui implique en retour que le rang de $\mathsf{S}^\star$ est plein. Si S est de taille (3×3) et interprétée comme une homographie agissant sur des points de $\mathbb{P}^2$, l'homographie duale, agissant sur les droites est $\mathsf{S}^\star$, ce qui se traduit par :

$$(\mathsf{S}\mathbf{x}) \wedge (\mathsf{S}\mathbf{y}) = \mathsf{S}^\star(\mathbf{x} \wedge \mathbf{y}). \tag{2.2}$$

[1] Nous utilisons les termes "Euclidien" et "métrique", en dépit du fait que "métrique" évoque la notion d'une mesure de distance "exacte" (c'est à dire pas à une échelle près), afin de reste consistant avec les termes Anglais "Euclidean" et "metric" utilisés, par exemple, par Hartley et Zisserman [93].

espace	nom de la transformation	modélisation	contraintes	# ddl	invariants
Euclidien	transformation rigide	$\mathsf{D} \sim \begin{pmatrix} & \mathsf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\mathsf{R} \in SO(3)$	6	longueurs, aires
métrique	similitude	$\mathsf{S} \sim \begin{pmatrix} & s\mathsf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\mathsf{R} \in SO(3)$	7	angles, rapports de longueur, points circulaires, conique absolue
affine	affinité	$\mathsf{A} \sim \begin{pmatrix} & \bar{\mathsf{A}} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix}$	-	12	parallélisme, centre de gravité, plan à l'infini, rapports d'aire
projectif	homographie	$\mathsf{H} \sim \begin{pmatrix} & \bar{\mathsf{H}} & & \mathbf{h}_1 \\ & \mathbf{h}_2^\mathsf{T} & & h \end{pmatrix}$	-	15	birapport, colinéarité, intersection, incidence

TAB. 2.1 – Caractérisation des *transformations* (aussi appelées *mouvements* ou *changements de base*) dans différents espaces. Les invariants des espaces les plus généraux sont valables dans les espaces moins généraux. Par exemple, le birapport est invariant pour tous les espaces mentionnés ci-dessus. "# ddl" indique le nombre de degrés de liberté de la transformation.

Cette relation peut être illustrée en dimension 3. Si $\mathbf{x}$ et $\mathbf{y}$ sont les coordonnées de points du plan projectif, la partie gauche de l'équation s'interprète comme la droite formée après avoir transféré les points par l'homographie S, et la partie droite comme le transfert de la droite formée avec les points originaux. Il en découle la formule :

$$\mathsf{S}[\mathbf{x}]_\wedge \mathsf{S}^\mathsf{T} \quad = \quad [\mathsf{S}^\star \mathbf{x}]_\wedge. \tag{2.3}$$

Les deux propriétés suivantes dépendent de la taille de la matrice S :

$$(\lambda \mathsf{S}_{(l\times l)})^\star \quad = \quad \lambda^{l-1} \mathsf{S}^\star \tag{2.4}$$

$$\det(\mathsf{S}^\star_{(l\times l)}) \quad = \quad (\det(\mathsf{S}))^{l-1}. \tag{2.5}$$

Une autre propriété est :

$$\mathsf{S}^\star \mathsf{S}^\mathsf{T} \quad = \quad \mathsf{S}^\mathsf{T} \mathsf{S}^\star \quad = \quad \det(\mathsf{S}) \mathrm{I}. \tag{2.6}$$

Elle est triviale à démontrer en utilisant l'expression (2.1) de S. Elle implique la propriété suivante :

$$(\det(\mathsf{S}) = 0) \quad \Rightarrow \quad \mathsf{S}^\star \mathsf{S}^\mathsf{T} = 0. \tag{2.7}$$

On montre aussi que :

$$(\mathrm{rang}(\mathsf{S}_{(l\times l)}) = l) \quad \Rightarrow \quad \mathrm{rang}(\mathsf{S}^\star) = l \tag{2.8}$$

$$(\mathrm{rang}(\mathsf{S}_{(l\times l)}) = l-1) \quad \Rightarrow \quad \mathrm{rang}(\mathsf{S}^\star) = 1 \tag{2.9}$$

$$(\mathrm{rang}(\mathsf{S}_{(l\times l)}) < l-1) \quad \Rightarrow \quad \mathrm{rang}(\mathsf{S}^\star) = 0. \tag{2.10}$$

Il découle des propriétés (2.7) et (2.9) que lorsque $\mathrm{rang}(\mathsf{S}_{(l\times l)}) = l-1$, S est définie de manière unique (à un facteur près) comme le noyau de $\mathsf{S}^{\mathsf{T}\star}$.

Pour terminer, lorsque $\det(\mathsf{S}) \neq 0$, l'opération réciproque de * est déduite de l'équation (2.1) :

$$(\mathsf{X} = \mathsf{S}^\star) \quad \Rightarrow \quad \mathsf{S} = \pm \mathsf{X}^\star / \sqrt{|\det(\mathsf{X})|}, \tag{2.11}$$

ce qui implique la relation :

$$\mathsf{S} \quad = \quad \pm (\mathsf{S}^\star)^\star / \sqrt{|\det(\mathsf{S}^\star)|}. \tag{2.12}$$

On en conclut que si S est de rang plein, $\mathsf{S}^\star$ encapsule S au signe près, lorsque S a une déficience de rang de 1, $\mathsf{S}^\star$ encapsule S à un facteur près, et lorsque S a une déficience de rang supérieure à 1, $\mathsf{S}^\star$ est la matrice nulle.

2.4 Quelques décompositions de matrices

Nous donnons brièvement les propriétés de quelques décompositions de matrices. Ces décompositions sont utilisées dans cette thèse pour la résolution de systèmes linéaires ou la paramétrisation de certaines entités géométriques. Nous présentons la *décomposition en valeurs singulières* puis la *décomposition* QR. Nous renvoyons par exemple au livre de Golub et Van Loan [71] pour une vue plus globale et de nombreux détails sur différentes décompositions de matrices. Notons que selon les auteurs ou la décomposition considérée, le terme factorisation est parfois utilisé (par exemple Golub et Van Loan parlent de la factorisation QR).

2.4.1 Décomposition en valeurs singulières

La *décomposition en valeurs singulières* ("Singular Value Decomposition" ou SVD en Anglais) est un outil permettant, entre autres, d'obtenir la solution d'un système linéaire exact, la solution au sens des moindres carrés d'un système linéaire sur-contraint et donne une famille de solutions pour un système linéaire sous-contraint. La décomposition en valeurs singulières donne aussi un diagnostique de la situation. Dans cette thèse, elle est principalement utilisée pour la résolution au sens des moindres carrés de systèmes linéaires sur-contraints. Nous ne la présentons que succinctement.

La décomposition en valeurs singulières est basée sur le théorème d'algébre linéaire suivant. Toute matrice A de taille $(r \times c)$ peut être décomposée comme suit :

$$\mathsf{A}_{(r\times c)} = \mathsf{U}_{(r\times c)}\mathsf{\Sigma}_{(c\times c)}\mathsf{V}^{\mathsf{T}}_{(c\times c)},$$

où U a des colonnes orthonormales, V est orthonormale et contient les vecteurs singuliers de A, et $\mathsf{\Sigma}$ est diagonale et contient les valeurs singulières de A. La plupart des algorithmes de décomposition en valeurs singulières ordonnent les valeurs singulières de manière décroissante. Lorsque le rang de A n'est pas plein, les dernières colonnes de V forment une base orthonormale de l'espace nul droit de A. Supposons $r \geq c$, comme c'est le cas lorsque A représente un système linéaire exact ou sur-contraint. Notons $a = \text{rang}(\mathsf{A}) \leq c$ le rang de A. La déficience de rang de A, c'est à dire la dimension de son espace nul droit, est donnée par $g = c - a$. On partitionne la décomposition en valeurs singulières comme :

$$\begin{aligned}\mathsf{\Sigma}_{(c\times c)} &= \text{diag}(\sigma_1, \ldots, \sigma_a, \underbrace{0, \ldots, 0}_{g}) \\ \mathsf{V}_{(c\times c)} &= \begin{pmatrix}\tilde{\mathsf{V}}_{(c\times a)} & \bar{\mathsf{V}}_{(c\times g)}\end{pmatrix}.\end{aligned}$$

Les colonnes de la matrice $\bar{\mathsf{V}}$ forment une base orthonormale de l'espace nul droit de A, c'est à dire $\mathsf{A}\bar{\mathsf{V}} = 0$.

2.4.2 Décomposition QR

La *décomposition* QR est basée sur le théorème suivant. Toute matrice A de taille $(r \times c)$ peut être décomposée comme suit :

$$\mathsf{A}_{(r\times c)} = \mathsf{Q}_{(r\times r)}\mathsf{R}_{(r\times c)},$$

où Q est orthonormale et R est triangulaire supérieure. Lorsque A est rectangulaire, R est constituée d'une sous-matrice triangulaire supérieure et, si $r > c$, complétée avec des lignes de zéros ou, si $r < c$, amputée de ses dernières lignes. Lorsque le rang de A n'est pas plein, un nombre de lignes de R correspondant à la déficience de rang est mis à zéro. Les dernières colonnes de Q forment une base de l'espace nul gauche de A. Par exemple, si $r < c$, avec les notations de la section précédente, on peut écrire :

$$\begin{aligned}\mathsf{R}_{(r\times c)} &= \begin{pmatrix}\tilde{\mathsf{R}}_{(a\times c)} \\ 0_{((r-a)\times c)}\end{pmatrix} \\ \mathsf{Q}_{(r\times r)} &= \begin{pmatrix}\tilde{\mathsf{Q}}_{(r\times a)} & \bar{\mathsf{Q}}_{(r\times g)}\end{pmatrix},\end{aligned}$$

où $\tilde{\mathsf{R}}_{(a\times c)}$ est une matrice triangulaire supérieure de taille $(c\times c)$ amputée de ses g dernières lignes. Les colonnes de la matrice $\bar{\mathsf{Q}}$ forment une base de l'espace nul gauche de A, c'est à dire $\bar{\mathsf{Q}}^\mathsf{T}\mathsf{A} = 0$.

La décomposition QL d'une matrice est similaire à la décomposition QR, à la différence près qu'un produit entre une matrice orthonormale et une matrice triangulation inférieure est obtenu. Il existe d'autres décompositions du même style : RQ, LQ, QP, TZ.

2.5 Mesures de distance

Nous examinons plusieurs mesures de distance entre entités géométriques (points, droites, plans, etc.), leurs fomulations algébriques et leurs caractéristiques. Ces mesures de distance sont utilisées tout au long de cette thèse dans les algorithmes d'estimation de transformation géométrique (par exemple de la matrice fondamentale). La mesure de distance "naturelle" entre entités géometriques est la *distance Euclidienne*, dont l'expression algébrique est souvent non-linéaire lorsque des coordonnées homogènes sont utilisées. De plus, certaines entités manipulées peuvent ne pas être de nature finie, en particulier lors de l'utilisation d'espaces projectifs, ce qui empêche l'utilisation d'une distance Euclidienne. Pour ces raisons, nous passons en revue des mesures de *distance algébriques*, basées sur une comparaison directe des vecteurs de coordonnées homogènes. Ces distances sont biaisées par rapport à la distance Euclidienne et n'ont pas de signification géométrique physique. Leur expression algébrique est souvent bilinéaire, et permet de construire des systèmes linéaires, par exemple l'algorithme des huit points pour l'estimation de la matrice fondamentale. Dans certains cas, le biais peut être exprimé explicitement via un *facteur de biais*. Certains algorithmes d'estimation utilisent cette propriété via un schéma de moindres carrés linéaires repondérés itérativement. Ces algorithmes sont souvent appelés *quasi-linéaires*. L'estimation d'entités géométriques est étudiée dans la section suivante.

Nous commençons par examiner les distances Euclidiennes puis algébriques, et enfin les facteurs de biais.

2.5.1 Distances Euclidiennes

Nous introduisons les distances Euclidiennes entre différentes entités et en différentes dimensions. Nous formulons ces distances en termes de coordonnées homogènes. Notons que la distance Euclidienne n'est définie qu'entre un point (variété de dimension 1) et une hypersurface (variété de dimension comprise entre 1 et $p-1$). Par exemple, on peut mesurer une distance Euclidienne entre un point 3D et un plan, entre un point 3D et une droite 3D ou entre deux points 3D. Les distances Euclidiennes ne sont définies qu'en espaces métrique et Euclidien.

Distance point-point. La distance entre deux points $\boldsymbol{x}$ et $\boldsymbol{y}$ de l'espace Euclidien de dimension p, représentés respectivement par des vecteurs de coordonnées homogènes de taille $((p+1)\times 1)$ notés $\mathbf{x}^\mathsf{T} \sim (\bar{\mathbf{x}}^\mathsf{T}\ \ x)$ et $\mathbf{y}^\mathsf{T} \sim (\bar{\mathbf{y}}^\mathsf{T}\ \ y)$, avec $x \neq 0$ et $y \neq 0$, est donnée par :

$$d^2_{PP}(\mathbf{x},\mathbf{y}) \quad = \quad \|\frac{\bar{\mathbf{x}}}{x} - \frac{\bar{\mathbf{y}}}{y}\|^2. \tag{2.13}$$

Distance point-hyperplan. La distance entre un point $\boldsymbol{x}$ et un hyperplan $\boldsymbol{\Pi}$ (par exemple une droite si $p = 2$ ou un plan si $p = 3$), représentés respectivement par des vecteurs de coordonnées homogènes de taille $((p+1)\times 1)$ notés $\mathbf{x}$ et $\mathbf{\Pi}$, avec $x \neq 0$ et $\|\bar{\mathbf{\Pi}}\| \neq 0$, est donnée par :

$$d^2_{PH}(\mathbf{x},\mathbf{\Pi}) \quad = \quad \frac{(\mathbf{x}^\mathsf{T}\mathbf{\Pi})^2}{x^2\|\bar{\mathbf{\Pi}}\|^2}. \tag{2.14}$$

Autres. Une distance Euclidienne peut aussi être calculée entre des points et des variétés linéaires ou non-linéaires de dimension quelconque. Il n'existe pas forcement de formulation analytique pour ces distances, comme par exemple pour la distance entre un point et une conique dans le plan Euclidien.

2.5.2 Distances algébriques

Nous considérons deux types de distances algébriques. Le premier concerne la comparaison d'entités de même dimension, par exemple deux points ou deux droites. Il est induit par la comparaison directe des vecteurs de coordonnées homogènes. Le deuxième type concerne la distance entre entités de dimensions différentes, par exemple entre un point et un hyperplan, et se formule à partir des contraintes algébriques entre les vecteurs de coordonnées homogènes. Ces distances algébriques n'ont pas de signification géométrique physique.

Distance vecteur-vecteur. Considérons les vecteurs de coordonnées homogènes $\mathbf{x}$ et $\mathbf{y}$ de taille $((p+1)\times 1)$. L'égalité des entités géométriques représentées par ces vecteurs est traduite par l'égalité à un facteur près de ces derniers, $\mathbf{x} \sim \mathbf{y}$. Cette dernière est équivalente à l'égalité des rapports entre les différents coefficients :

$$\begin{aligned}(\mathbf{x} \sim \mathbf{y}) &\Leftrightarrow \frac{x_k}{x_l} = \frac{y_k}{y_l}, \quad k = 1 \dots p+1,\ l = 1 \dots p+1,\ l > k \\ &\Leftrightarrow x_k y_l = x_l y_k, \quad k = 1 \dots p+1,\ l = 1 \dots p+1,\ l > k.\end{aligned}$$

Parmi ces $p(p+1)/2$ équations, seules p sont indépendantes. Cependant, on ne peut pas en sélectionner p à priori dans le cas général, car si un coefficient s'annule, les équations incluant ce coefficient dégénèrent. Par exemple si $x_1 = 0$, toutes les équations avec $k = 1$ ou $l = 1$ dégénèrent. Nous formulons donc la distance algébrique entre deux vecteurs de coordonnées homogènes par la formulation bilinéaire suivante :

$$d^2_{A_VV}(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{p+1} \sum_{l=1, l>k}^{p+1} (x_k y_l - x_l y_k)^2.$$

Une autre manière de formuler cette distance est de considérer la matrice D de taille $((p+1)\times(p+1))$ définie par :

$$\mathsf{D} = \mathbf{x}\mathbf{y}^\mathsf{T} - \mathbf{y}\mathbf{x}^\mathsf{T}.$$

Les coefficients de cette matrice correspondent aux équations ci-dessus et s'annulent lorsque les vecteurs $\mathbf{x}$ et $\mathbf{y}$ sont égaux à un facteur près. La norme de cette matrice est proportionnelle à la distance algébrique entre $\mathbf{x}$ et $\mathbf{y}$:

$$d^2_{A_VV}(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x}\mathbf{y}^\mathsf{T} - \mathbf{y}\mathbf{x}^\mathsf{T}\|^2. \tag{2.15}$$

La mesure de distance algébrique définie ci-dessus concerne deux vecteurs de coordonnées homogènes. Ces derniers peuvent représenter des points, des droites, des plans ou n'importe quelles autres entités géométriques, finies ou infinies. Par exemple, nous l'utilisons au chapitre 8 pour comparer les coordonnées de Plücker de droites 3D. Elle n'a pas de sens géométrique physique. En particulier, sa valeur dépend du facteur d'échelle libre de ses arguments. En dimension deux, c'est à dire dans le plan projectif, nous obtenons la formule suivante entre les vecteurs-3 $\mathbf{m}$ et $\mathbf{n}$:

$$d^2_{A_VV}(\mathbf{m}, \mathbf{n}) = \|\mathbf{m} \wedge \mathbf{n}\|^2. \tag{2.16}$$

Cas de points de l'espace Euclidien. Considérons le cas où les arguments $\mathbf{x}$ et $\mathbf{y}$ de la distance algébrique représentent des points $\boldsymbol{x}$ et $\boldsymbol{y}$ de espace Euclidien, c'est à dire $x \neq 0$ et $y \neq 0$. Notons qu'on ne peut pas poser de contraintes similaires sur les coordonnées homogènes des hyperplans ou autres variétés. Dans ce cas, on sait que les p équations impliquant x et y ne dégénèrent pas. On peut donc construire un système de p équations minimal :

$$d^2_{A_PP}(\mathbf{x}, \mathbf{y}) = \|x\bar{\mathbf{y}} - y\bar{\mathbf{x}}\|^2. \tag{2.17}$$

En dimension 2, c'est à dire dans le plan Euclidien, nous obtenons la formule suivante pour la distance algébrique entre les points $\boldsymbol{m}$ et $\boldsymbol{n}$ avec $m_3 \neq 0$ et $n_3 \neq 0$:

$$d^2_{A_PP}(\mathbf{m},\mathbf{n}) \;=\; \|\mathsf{S}(\mathbf{m}\wedge\mathbf{n})\|^2 \quad \text{avec} \quad \mathsf{S} \;=\; \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{2.18}$$

Cette formule sera utilisée entre autres lors de la comparaison de points mesurés dans une image et de points prédits par un modèle 3D.

Distance point-hyperplan. La condition algébrique pour qu'un point de coordonnées $\mathbf{x}$ soit sur un hyperplan de coordonnées $\mathbf{\Pi}$ (par exemple en dimension 2, pour qu'un point soit sur une droite) est $\mathbf{x}^\mathsf{T}\mathbf{\Pi} = 0$. Cette condition induit la distance algébrique suivante :

$$d^2_{A_PH}(\mathbf{x},\mathbf{\Pi}) \;=\; (\mathbf{x}^\mathsf{T}\mathbf{\Pi})^2. \tag{2.19}$$

2.5.3 Relier les distances Euclidiennes et algébriques

Les distances Euclidienne et algébrique sont différentes. On dit que la distance algébrique est biaisée par rapport à la distance Euclidienne, qui elle, a un sens physique. Le *facteur de biais* reliant les distances Euclidienne et algébrique est formulé explicitement ci-dessous, pour les deux types de distances Euclidienne données ci-dessus : les distances point-point et point-hyperplan de l'espace Euclidien.

Distance point-point. Examinons les expressions (2.13) et (2.17) des distances Euclidienne et algébrique entre deux points $\boldsymbol{x}$ et $\boldsymbol{y}$ de l'espace Euclidien :

$$\begin{aligned} d^2_{PP}(\mathbf{x},\mathbf{y}) &= \|\frac{\bar{\mathbf{x}}}{x} - \frac{\bar{\mathbf{y}}}{y}\|^2 \\ d^2_{A_PP}(\mathbf{x},\mathbf{y}) &= \|x\bar{\mathbf{y}} - y\bar{\mathbf{x}}\|^2. \end{aligned}$$

Il est trivial de voir que si l'on multiplie la distance Euclidienne par xy, on obtient la distance algébrique, d'où la définition suivante du *facteur de biais* noté w_{PP} :

$$d^2_{PP}(\mathbf{x},\mathbf{y}) \;=\; w^2_{PP}\, d^2_{A_PP}(\mathbf{x},\mathbf{y}) \tag{2.20}$$

$$w^2_{PP}(\mathbf{x},\mathbf{y}) \;=\; \frac{1}{x^2y^2}. \tag{2.21}$$

Distance point-hyperplan. Examinons les expressions (2.14) et (2.19) des distances Euclidienne et algébrique entre un point $\boldsymbol{x}$ et un hyperplan $\boldsymbol{\Pi}$ de l'espace Euclidien :

$$\begin{aligned} d^2_{PH}(\mathbf{x},\mathbf{\Pi}) &= \frac{(\mathbf{x}^\mathsf{T}\mathbf{\Pi})^2}{x^2\|\bar{\mathbf{\Pi}}\|^2} \\ d^2_{A_PH}(\mathbf{x},\mathbf{\Pi}) &= (\mathbf{x}^\mathsf{T}\mathbf{\Pi})^2. \end{aligned}$$

Similairement au cas précédent, ces deux distances sont reliées par un *facteur de biais* noté w_{PH} et défini par :

$$d^2_{PH}(\mathbf{x},\mathbf{\Pi}) \;=\; w^2_{PH}\, d^2_{A_PH}(\mathbf{x},\mathbf{\Pi}) \tag{2.22}$$

$$w^2_{PH}(\mathbf{x},\mathbf{\Pi}) \;=\; \frac{1}{x^2\|\bar{\mathbf{\Pi}}\|^2}. \tag{2.23}$$

2.6 Estimation d'entités géométriques, optimisation linéaire et quasi-linéaire

L'estimation d'entités géométriques est un sujet très général englobant de nombreux problèmes abordés dans cette thèse, et plus généralement en vision par ordinateur, tels la triangulation à partir d'images ou le calcul de la matrice fondamentale. La majorité des algorithmes d'estimation sont basés sur la comparaison de primitives géométriques mesurées et de primitives prédites par un modèle. Les coordonnées homogènes des primitives prédites s'expriment souvent de manière linéaire en termes des inconnues. Les transformations peuvent être estimées par la résolution au sens des moindres carrés de systèmes linéaires. L'erreur minimisée, en d'autres termes la fonction de coût, s'écrit comme une somme de distances algébriques entre les primitives mesurées et les primitives prédites.

La minimisation d'une fonction de coût basée sur une distance algébrique n'est pas forcement une bonne méthode. Elle est souvent utilisée pour initialiser la minimisation itérative d'une fonction de coût non-linéaire basée sur une distance Euclidienne. Des techniques de minimisation non-linéaire générales sont présentés au chapitre 3 dans le cadre de l'ajustement de faisceaux. Nous présentons dans cette section une technique itérative plus spécifique dite *quasi-linéaire*. Cette technique est basée sur le fait qu'une distance Euclidienne est reliée à une distance algébrique par un facteur de biais. L'optimisation quasi-linéaire utilise un schéma de moindres carrés itérativement repondérés, permettant l'estimation des facteurs de biais.

Afin d'expliquer de manière concrète les différents concepts présentés dans cette section, nous les illustrons sur l'exemple de l'estimation d'une homographie 2D à partir de correspondances de points image. Cette idée provient du chapitre 3 du livre de Hartley et Zisserman [93]. Considérons un ensemble de m correspondances de points mesurés $\boldsymbol{q}_j \leftrightarrow \boldsymbol{q}'_j$, reliés par une homographie 2D représentée par la matrice (3×3) H, c'est à dire que si les mesures de ces points n'étaient pas bruitées, on aurait $\mathbf{q}'_j \sim \mathsf{H}\mathbf{q}_j$. Sauf mention contraire, nous supposons que ces correspondances donnent suffisamment de contraintes pour définir une unique homographie 2D : au moins 4 correspondances sont définies, et les points de chaque image définissent une base du plan projectif (aucun sous-ensemble de $m-1$ points n'est colinéaire).

Notons que le problème de l'estimation "optimale" d'une homographie, dans le sens du maximum de vraisemblance, rentre dans le cadre des algorithmes d'ajustement de faisceaux étudiés au chapitre 3, et n'est pas traité dans cette section.

Pour plus de détails sur la résolution de systèmes linéaires, nous renvoyons au livre de Golub et Van Loan [71] et [119, 156]. Cette section est organisée comme suit. Nous formulons un estimateur linéaire pour homographie 2D, basé sur la résolution au sens des moindres carrés d'un système linéaire homogène. Nous montrons ensuite comment une fonction de coût basée sur la distance Euclidienne peut être minimisée par optimisation quasi-linéaire. Finalement, nous présentons la résolution au sens des moindres carrés d'un système linéaire non homogène puis une technique d'optimisation linéaire sous contraintes linéaires.

2.6.1 Optimisation linéaire

Nous formulons un système linéaire pour l'estimation d'une homographie 2D puis donnons une technique de résolution au sens des moindres carrés basée sur la décomposition en valeurs singulières. Enfin, nous analysons la fonction de coût minimisée par cet algorithme.

2.6.1.1 Formulation du système linéaire

La formulation donnée ci-dessous est très proche de celle de Hartley et Zisserman [93, §3.1]. Les contraintes exploitées sont issues du fait que l'homographie H transfère un point sur son correspondant dans l'autre image : $\mathbf{q}'_j \sim \mathsf{H}\mathbf{q}_j$. L'égalité à un facteur près entre deux vecteurs de coordonnées homogènes implique que le produit vectoriel s'annule, ce qui donne les 3 équations $\mathbf{q}'_j \wedge (\mathsf{H}\mathbf{q}_j) = \mathbf{0}_{(3\times1)}$. Parmi ces 3 équations, seules 2 sont indépendantes. Dans le cas général, on ne peut pas sélectionner 2 équations a priori. Par exemple si $\boldsymbol{q}_j$ ou $\boldsymbol{q}'_j$ est à l'infini, alors les deux premières équations sont linéairement dépendantes. Par contre, lorsque des points image, c'est à dire du plan Euclidien, sont considérés, on peut omettre la troisième équation, et ne retenir que les deux premières :

$$\mathsf{S}\left(\mathbf{q}'_j \wedge (\mathsf{H}\mathbf{q}_j)\right) \quad = \quad \mathbf{0}_{(2\times1)},$$

où la matrice S, définie par l'équation (2.18), sert à sélectionner les deux premières équations. Réorganisons cette expression en termes d'un vecteur-9 $\mathbf{h}$ contenant les coefficients de H, et défini comme $\mathbf{h} = \text{vect}(\mathsf{H})$:

$$\mathsf{A}_j\mathbf{h} \;=\; \mathbf{0} \quad \text{avec} \quad \mathsf{A}_{j_{(2\times 9)}} \;=\; \begin{pmatrix} \mathbf{0}^\mathsf{T}_{(3\times 1)} & -q'_{j,3}\mathbf{q}_j^\mathsf{T} & q'_{j,2}\mathbf{q}_j^\mathsf{T} \\ q'_{j,3}\mathbf{q}_j^\mathsf{T} & \mathbf{0}^\mathsf{T}_{(3\times 1)} & -q'_{j,1}\mathbf{q}_j^\mathsf{T} \end{pmatrix},$$

et la notation $\mathbf{q}'^\mathsf{T}_j \sim (q'_{j,1}\;\; q'_{j,2}\;\; q'_{j,3})$. En empilant les équations obtenues pour chaque correspondance de points, on obtient le système linéaire suivant :

$$\mathsf{A}\mathbf{h} \;=\; \mathbf{0} \quad \text{avec} \quad \mathsf{A}^\mathsf{T}_{(2m\times 9)} \;=\; (\cdots\;\; \mathsf{A}_j^\mathsf{T}\;\; \cdots). \tag{2.24}$$

Un tel système, avec un membre droit nul, est appelé un *système homogène*, par opposition aux systèmes de la forme $\mathsf{A}\mathbf{x} = \mathbf{b}$ avec $\mathbf{b} \neq \mathbf{0}$. En l'absence de bruit sur la position des points, c'est à dire lorsque les points sont exactement mis en correspondance par une homographie 2D, la matrice A est de rang 8 et a un espace nul de dimension 1. Dans ce cas, une solution exacte pour H est donnée par un vecteur nul de A.

2.6.1.2 Résolution au sens des moindres carrés

Lorsque les points ne se correspondent pas exactement par une homographie 2D, le rang de la matrice A est plein. Elle n'a donc pas d'espace nul, ce qui reflète le fait qu'il n'existe pas de solution exacte pour H. On résout alors le système en minimisant un certain critère d'erreur. La résolution au sens des moindres carrés minimise le critère suivant :

$$\mathcal{C}_A \;=\; \|\mathsf{A}\mathbf{h}\|^2. \tag{2.25}$$

L'indice A indique que la fonction de coût est algébrique (voir ci-dessous). La matrice H étant définie à un facteur près, $\mathbf{h}$ est un vecteur de coordonnées homogènes, dont l'échelle est fixée par la contrainte $\|\mathbf{h}\|^2 = 1$. La solution minimisant $\mathcal{C}_A$ sous la contrainte $\|\mathbf{h}\|^2 = 1$ est donnée par le vecteur singulier associé à la plus petite valeur singulière de la matrice A. On peut calculer ce vecteur par la décomposition en valeurs singulières de A : $\mathsf{A} \sim \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$. La solution recherchée est donnée par la colonne de V associée à la plus petite valeur singulière, en pratique, la dernière colonne de V.

Notons que si une homographie 2D n'est pas déterminée de manière unique, c'est à dire si le nombre de correspondances de points est inférieur à 4, ou si les points ne sont pas en position générale, alors on obtient une famille de solutions. Cette famille est paramétrée par les "derniers" vecteurs singuliers de A (les vecteurs singuliers associés aux plus petites valeurs singulières). Le degré de dégénérescence détermine combien de vecteurs singuliers forment la famille de solutions. Par exemple, avec 3 correspondances de points, on obtient une famille de dimension 2, dont une base est fournie par les 3 "derniers" vecteurs singuliers de A. Si le degré de dégénérescence est inconnu a priori, il peut être déterminé en analysant les valeurs singulières de A. De manière plus générale, on arrive à un problème de choix de modèle. Nous n'abordons pas ce problème dans cette thèse et renvoyons aux travaux de Torr et al. [212] et de Kanatani [111].

2.6.1.3 Fonction de coût induite

Développons la fonction de coût (2.25) par un chemin inverse à la construction du système linéaire :

$$\mathcal{C}_A \;=\; \|\mathsf{A}\mathbf{h}\|^2 \;=\; \textstyle\sum_{j=1}^m \|\mathsf{A}_j\mathbf{h}\|^2 \;=\; \sum_{j=1}^m \|\mathsf{S}[\mathbf{q}'_j]_\wedge \mathsf{H}\mathbf{q}_j\|^2 \;=\; \sum_{j=1}^m d^2_{A_PP}(\mathbf{q}'_j, \mathsf{H}\mathbf{q}_j), \tag{2.26}$$

où $d^2_{A_PP}$ est la distance algébrique point-point donnée par l'équation (2.18). La résolution au sens des moindres carrés minimise donc la somme des carrés des distances algébriques entre les points de la deuxième image et les points de la première image transférés par l'homographie. De manière générale, un tel critère est appelé une *erreur de transfert non symétrique*. On peut faire plusieurs remarques par rapport à ce critère : (i) il n'est pas symétrique, dans le sens où les deux images n'ont pas le même rôle (l'erreur est minimisée dans la deuxième image seulement), et (ii) il est biaisé par rapport à un critère qui serait basé sur une distance Euclidienne pour comparer les points.

2.6.1.4 Normalisation des données

Hartley [86] montre qu'une simple normalisation des points image améliore énormement le calcul de la matrice fondamentale avec l'algorithme linéaire des huit points. Cette normalisation consiste à translater le centroïde des points image au centre du repère et à appliquer un facteur d'échelle tel que la distance moyenne des points à l'origine soit $\sqrt{2}$. De manière plus générale, cette normalisation améliore les résultats de la plupart des algorithmes linéaires, voir par exemple [93]. Hartley montre que la normalisation améliore le conditionnement numérique du système linéaire. Depuis, d'autres explications ont été proposées. En particulier, Mühlich et Mester [149] et Chojnacki et al. [41] montrent que la normalisation rend l'erreur algébrique minimisée par l'algorithme linéaire plus proche de l'erreur de reprojection (voir le chapitre 3).

Tout nos algorithmes comportent une normalisation des données, les algorithmes linéaires comme les algorithmes non-linéaires. Lorsque plusieurs images sont traitées, nous choisissons la même normalisation dans toutes les images, afin de ne pas modifier leurs contributions relatives au système d'équations.

2.6.2 Optimisation quasi-linéaire

Modifions la fonction de coût (2.25) afin de lui donner un sens physique. Pour ce faire, remplaçons la distance algébrique par la distance Euclidienne point-point dans l'équation (2.26). Ceci définit la fonction de coût suivante :

$$\mathcal{C} \quad = \quad \sum_{j=1}^{m} d_{PP}^2(\mathbf{q}'_j, \mathsf{H}\mathbf{q}_j).$$

Cette fonction de coût est non-linéaire. Tous les termes correspondants des fonctions de coût $\mathcal{C}$ et $\mathcal{C}_A$ sont reliés par un facteur de biais, dont la valeur dépend de la transformation géométrique inconnue. Les facteurs de biais sont formulés d'après l'équation (2.21) :

$$w_{PP}^2(\mathbf{q}'_j, \mathsf{H}\mathbf{q}_j) \quad = \quad \frac{1}{{q'_{j,3}}^2(\mathbf{h}_3^\mathsf{T}\mathbf{q}_j)^2}, \tag{2.27}$$

où $\mathbf{h}_3$ est la troisième ligne de la matrice H. Les deux fonctions de coût sont donc reliées par un ensemble de tels facteurs. Cela suggère un schéma de minimisation itératif, basé sur les moindres carrés linéaires itérativement repondérés. Une solution biaisée minimisant $\mathcal{C}_A$ est calculée, basée sur la résolution au sens des moindres carrés d'un système linéaire. A partir de cette solution, les facteurs de biais sont estimés, et le système linéaire est repondéré. Ces deux étapes sont itérées jusqu'à convergence. Ce type d'algorithme est souvent appelé *quasi-linéaire*. Dans le cadre des problèmes de vision par ordinateur, c'est à dire avec un bruit sur les données et des erreurs résiduelles peu élevées, la plupart des algorithmes quasi-linéaires convergent après 2 à 5 itérations. Les avantages sont la simplicité d'implantation par rapport à un algorithme plus général de type Newton nécessitant le calcul des dérivées de la fonction de coût (voir le chapitre 3, §3.6), et le temps de calcul nécessaire pour une itération, en général inférieur aux autres algorithmes minimisant une erreur Euclidienne. L'algorithme d'estimation quasi-linéaire d'une homographie 2D est résumé dans le tableau 2.2. La convergence peut être déterminée de différentes manières. Par exemple, on peut seuiller la distance entre deux vecteurs de facteurs de biais consécutifs, ou entre deux erreurs résiduelles consécutives.

L'optimisation quasi-linéaire est utilisée entre autres par Hartley [91] pour l'estimation du tenseur trifocal et par Demirdjian et Horaud [51] pour l'estimation d'une homographie 3D.

Quasi-linéarisation d'un algorithme linéaire. Comme illustré ci-dessus sur l'exemple de l'estimation d'une homographie 2D, un algorithme linéaire peut être "quasi-linéarisé", sous certaines conditions. Si il minimise une fonction de coût pouvant être écrite comme une somme de distances algébriques pour lesquelles il existe une distance Euclidienne équivalente, alors il peut être quasi-linéarisé. La fonction de coût minimisée par l'algorithme quasi-linéaire est obtenue en remplaçant la distance algébrique par la distance Euclidienne dans la fonction de coût de l'algorithme linéaire. Il existe une distance Euclidienne entre les points et les hyperplans de nature finie, par exemple entre un point et une droite image, et entre les points finis. Par contre, il n'existe

1. *Initialisation* : former le système linéaire représenté par la matrice $\mathsf{A} = \mathsf{A}_0$ donnée par l'équation (2.24), initialiser le compteur d'itération $k = 0$;
2. *Estimation* : résoudre le système au sens des moindres carrés par décomposition en valeurs singulières de A_k : $\min_{\mathbf{h}_k, \|\mathbf{h}_k\|^2=1} \|\mathsf{A}_k \mathbf{h}_k\|^2$;
3. *Correction du biais* : former A_{k+1} en repondérant A_k par les facteurs de biais définis par l'équation (2.27) ;
4. *Itération* : $k \rightarrow k+1$, itérer les étapes 2 et 3 jusqu'à convergence.

TAB. 2.2 – Algorithme d'estimation quasi-linéaire d'une homographie 2D à partir de correspondances de points image. Cet algorithme minimise la somme des distances Euclidiennes entre les points de l'image de droite et les points de l'image de gauche, transférés par l'homographie.

pas de distance Euclidienne entre deux hyperplans, alors que ces derniers peuvent être comparés par la distance algébrique d_{VV} donnée par l'équation (2.15).

2.6.3 Systèmes non homogènes

Certains problèmes examinés dans cette thèse conduisent à la résolution de systèmes non homogènes de la forme $\mathsf{A}\mathbf{x} = \mathbf{b}$ avec $\mathbf{b} \neq \mathbf{0}$. Dans le cas de l'estimation d'une homographie 2D, on peut en fixer l'échelle telle que l'un de ses coefficients, par exemple H_{33}, soit égal à 1. On obtient alors un système non homogène. Hartley et Zisserman [93, §3.1.2] montrent que cette modélisation peut conduire à des instabilités numériques si la solution pour H est telle que H_{33} est proche de 0.

Un système non homogène $\mathsf{A}\mathbf{x} = \mathbf{b}$ a une solution unique lorsque le rang de A est plein[2]. La solution au sens des moindres carrés minimise le critère d'erreur suivant :

$$\mathcal{C}_A = \|\mathsf{A}\mathbf{x} - \mathbf{b}\|^2.$$

Elle est donnée par la résolution des équations normales induites par le système :

$$\begin{aligned} \mathsf{A}^\mathsf{T}\mathsf{A}\mathbf{x} &= \mathsf{A}^\mathsf{T}\mathbf{b} \\ \mathbf{x} &= \underbrace{(\mathsf{A}^\mathsf{T}\mathsf{A})^{-1}\mathsf{A}^\mathsf{T}}_{\mathsf{A}^\dagger}\mathbf{b}, \end{aligned}$$

où $\mathsf{A}^\dagger$ est appelée une pseudo-inverse de A. Les méthodes d'optimisation présentées au chapitre 3 nécessitent la résolution d'équations normales. Dans certains cas, ces équations normales impliquent une matrice A dont le rang, noté a, n'est pas plein, ce qui implique que le rang de $\mathsf{A}^\mathsf{T}\mathsf{A}$ n'est pas plein non plus. Si $(r \times c)$ est la taille de A, alors la déficience de rang est donnée par $g = c - a$. On obtient une famille de solutions, paramétrée par un vecteur $\mathbf{v}$ de taille $(g \times 1)$:

$$\mathbf{x} = \mathsf{A}^\ddagger\mathbf{b} + \Lambda\mathbf{v}.$$

Dans cette équation, Λ est une matrice de taille $(c \times g)$ dont les colonnes forment une base de l'espace nul de A. La notation $\mathsf{X}^\ddagger$ désigne la pseudo-inverse de Moore-Penrose de la matrice carrée X. Elle se calcule comme suit. Soit $\mathsf{X} = \mathsf{U}\,\mathrm{diag}(\sigma_1, \ldots, \sigma_c)\,\mathsf{V}^\mathsf{T}$ une décomposition en valeurs singulières, $\mathsf{X}^\ddagger$ est définie par $\mathsf{X}^\ddagger = \mathsf{V}\,\mathrm{diag}(\tilde{\sigma}_1, \ldots, \tilde{\sigma}_c)\,\mathsf{U}^\mathsf{T}$, où $\tilde{\sigma}_i$ est donné par $1/\sigma_i$ si $\sigma_i \neq 0$ et 0 sinon.

La méthode de choix pour la résolution au sens des moindres carrés des systèmes linéaires non homogènes est, comme dans le cas des systèmes linéaires homogènes, la décomposition en valeurs singulières, car elle permet de traiter les cas exacts, sur-contraints et sous-contraints et donne une analyse de la situation au travers des valeurs singulières.

Comme dans le cas des systèmes homogènes, on peut appliquer un schéma d'estimation quasi-linéaire en repondérant itérativement la matrice A et le vecteur $\mathbf{b}$ par les facteurs de biais.

[2] Comme on suppose que la matrice A a au moins autant de lignes que de colonnes, le rang de A correspond au rang sur les colonnes.

2.6.4 Optimisation sous contraintes linéaires

Le problème de la résolution au sens des moindres carrés d'un système linéaire homogène sous contraintes linéaires homogènes est formulé par :

$$\min_{\mathbf{x},\|\mathbf{x}\|^2=1,\mathsf{C}\mathbf{x}=\mathbf{0}} \|\mathsf{A}\mathbf{x}\|^2,$$

où A est une matrice $(r \times c)$ et C une matrice $(s \times c)$ modélisant les s contraintes linéaires. Nous supposons que le nombre de contraintes s est inférieur au nombre d'inconnues : $s < c$. La transposition des méthodes aux cas d'un système non homogène et / ou de contraintes non homogènes est triviale et ne sera pas donnée explicitement. Il existe plusieurs méthodes de résolution, entre autres, les méthodes par élimination, les méthodes par pondération et les multiplicateurs de Lagrange. Nous renvoyons aux livres cités plus haut pour plus de détails.

2.6.4.1 Une méthode par pondération

Les méthodes par pondération fournissent une solution approchée du problème. Elles sont basées sur l'introduction des contraintes au système original, affectées d'un poids w :

$$\begin{pmatrix} \mathsf{A} \\ w\mathsf{C} \end{pmatrix} \mathbf{x} = \mathbf{0}.$$

Notons qu'il est possible d'utiliser un poid différent pour chaque contrainte, en multipliant la matrice C par une matrice diagonale $(s \times s)$. La résolution au sens des moindres carrés de ce système minimise la fonction de coût suivante :

$$\min_{\mathbf{x},\|\mathbf{x}\|^2=1} \|\mathsf{A}\mathbf{x}\|^2 + w^2\|\mathsf{C}\mathbf{x}\|^2.$$

On montre que la solution obtenue tend vers la solution recherchée lorsque w tend vers l'infini. De grandes valeurs numériques pour w assurent que les contraintes sont satisfaites au delà de la précision numérique utilisée. L'avantage de cette approche est qu'elle est très simple. Le problème est que de grandes valeurs pour w déséquilibrent la résolution numérique du système. Il y a donc un compromis à trouver pour la valeur de w, entre la stabilité (le conditionnement numérique) du système linéaire et la précision de satisfaction des contraintes. Le nombre d'inconnues n'est pas réduit par la présence des contraintes, et le nombre d'équations est augmenté. Pour ces raisons, nous préférons la méthode par élimination présentée ci-dessous.

2.6.4.2 Une méthode par élimination

Pour simplifier les notations, on suppose sans perte de généralité que A et C sont de rang plein. L'idée est de calculer un changement de base orthonormal de l'espace des inconnues tel que les contraintes s'écrivent de manière triviale dans cette nouvelle base, et plus précisément qu'elles soient équivalentes à annuler certaines inconnues. Il existe plusieurs solutions pour calculer une telle transformation. Considérons par exemple la décomposition QR suivante :

$$\mathsf{C}^\mathsf{T}_{(s\times c)} = \mathsf{Q}_{(c\times c)}\mathsf{R}_{(c\times s)},$$

où Q est une matrice orthonormale et R est partitionnée en une matrice triangulaire supérieure $\tilde{\mathsf{R}}_{(s\times s)}$ et des lignes de zéros, voir §2.4.2 :

$$\mathsf{R}_{(c\times s)} = \begin{pmatrix} \tilde{\mathsf{R}}_{(s\times s)} \\ \mathsf{0}_{((c-s)\times s)} \end{pmatrix}.$$

La matrice Q^T représente le changement de base recherché. En effet, les contraintes $\mathsf{C}\mathbf{x} = \mathbf{0}$ se reécrivent $\mathsf{R}^\mathsf{T}\mathsf{Q}^\mathsf{T}\mathbf{x} = \mathbf{0}$. On pose $\mathbf{y} = \mathsf{Q}^\mathsf{T}\mathbf{x}$. Si l'on partitionne $\mathbf{y}$ comme $\mathbf{y}^\mathsf{T}_{(c\times 1)} = (\tilde{\mathbf{y}}^\mathsf{T}_{(s\times 1)} \;\; \bar{\mathbf{y}}^\mathsf{T}_{((c-s)\times 1)})$, les contraintes se

reécrivent :

$$\mathsf{C}\mathbf{x} \;=\; \begin{pmatrix}\tilde{\mathsf{R}}^{\mathsf{T}} & 0\end{pmatrix}\begin{pmatrix}\tilde{\mathbf{y}}\\ \bar{\mathbf{y}}\end{pmatrix} \;=\; \tilde{\mathsf{R}}^{\mathsf{T}}\tilde{\mathbf{y}} \;=\; \mathbf{0},$$

d'où l'on déduit que les contraintes sont satisfaites si et seulement si $\tilde{\mathbf{y}} = \mathbf{0}$.

On peut aussi utiliser une décomposition en valeurs singulières pour calculer un changement de base approprié : $\mathsf{C}_{(s\times c)} = \mathsf{U}_{(s\times c)}\Sigma_{(c\times c)}\mathsf{V}^{\mathsf{T}}_{(c\times c)}$. La matrice V joue un rôle similaire à Q.

L'ensemble des vecteurs $\mathbf{x}$ satisfaisant les contraintes est donc paramétré par :

$$\mathbf{x} \;=\; \mathsf{Q}\mathbf{y} \;=\; \begin{pmatrix}\tilde{\mathsf{Q}} & \bar{\mathsf{Q}}\end{pmatrix}\begin{pmatrix}\mathbf{0}_{(s\times 1)}\\ \bar{\mathbf{y}}\end{pmatrix} \;=\; \bar{\mathsf{Q}}\bar{\mathbf{y}}.$$

Substituons cette paramétrisation dans le problème original :

$$\arg\min_{\mathbf{x},\|\mathbf{x}\|^2=1,\mathsf{C}\mathbf{x}=\mathbf{0}} \|\mathsf{A}\mathbf{x}\|^2 \;=\; \bar{\mathsf{Q}}\left(\arg\min_{\bar{\mathbf{y}},\|\bar{\mathbf{y}}\|^2=1} \|\bar{\mathsf{A}}\bar{\mathbf{y}}\|^2\right) \quad \text{avec} \quad \bar{\mathsf{A}} \;=\; \mathsf{A}\bar{\mathsf{Q}}.$$

Notons que l'égalité $\|\mathbf{x}\|^2 = \|\bar{\mathsf{Q}}\bar{\mathbf{y}}\|^2 = \|\bar{\mathbf{y}}\|^2$ provient du fait que Q ait des colonnes orthonormales. Le problème original est donc transformé en un problème linéaire non contraint de taille inférieure, que l'on peut résoudre par exemple par une décomposition en valeurs singulières de la matrice $\bar{\mathsf{A}}$.

Hartley et Zisserman [93, §A3.4.3] décrivent cette méthode où la décomposition en valeurs singulières de C est utilisée pour trouver le changement de base. Une technique d'optimisation sous contrainte peut évidemment être utilisé dans un schéma d'optimisation quasi-linéaire. On peut même profiter de la nature itérative de l'optimisation quasi-linéaire pour linéariser des contraintes non-linéaires autour de l'estimation courante et les inclure au système. Nous utilisons une telle idée au chapitre 6 pour la triangulation multi-vues de droites.

2.7 Matrices orthonormales et de rotation

Cette section est dédiée à la présentation du groupe de Lie des *matrices orthonormales* de dimension p noté $O(p)$ et sa spécialisation aux *matrices de rotation* noté $SO(p)$. Ces matrices sont utilisées pour représenter les rotations dans un espace de dimension p. Elles ont $p(p-1)/2$ degrés de liberté. Par exemple en dimension $p = 3$, les 3 degrés de liberté peuvent être interprétés comme 3 angles de rotation (ce sont les angles d'Euler, voir §2.7.2). Ces matrices sont aussi à la base des représentations orthonormales que nous proposons, au chapitre 4 pour représenter la matrice fondamentale, et au chapitre 6 pour représenter les droites 3D. Il existe d'autres représentations des rotations 3D (quaternion unitaire, vecteur de Gibbs, paramètres de Cayley-Klein, l'axe et l'angle, etc.). Nous renvoyons au livre de Faugeras [58] et aux travaux de Stuelpnagel [192] pour plus de détails. Les matrices orthonormales possèdent des propriétés particulières. Il n'existe pas de paramétrisation minimale, complète et linéaire de leurs $p(p-1)/2$ degrés de liberté, voir par exemple [58, §5.5].

Nous définissons les groupes $O(p)$ et $SO(p)$ puis présentons la décomposition des matrices de rotation en angles élémentaires et un schéma d'estimation non-linéaire.

2.7.1 Définition

Les *matrices orthonormales* sont soumises aux *contraintes d'orthonormalité* :

$$\mathsf{R} \in O(p) \quad \Leftrightarrow \quad \mathsf{R}^{\mathsf{T}}\mathsf{R} = \mathrm{I}_{(p\times p)}. \tag{2.28}$$

Elles ont un déterminant dont la valeur absolue est 1. Un sous-groupe est construit avec les matrices dont le déterminant est positif, c'est à dire dont la valeur est 1. Ce sont les *matrices de rotation* :

$$\mathsf{R} \in SO(p) \quad \Leftrightarrow \quad \mathsf{R}^{\mathsf{T}}\mathsf{R} = \mathrm{I}_{(p\times p)},\ \det(\mathsf{R}) = 1. \tag{2.29}$$

Les matrices orthonormales préservent, entre autres, la norme $\mathcal{L}_2$:

$$\mathsf{R} \in O(p) \quad \Leftrightarrow \quad \forall \mathbf{x} \in \mathbb{R}^p,\ \|\mathbf{x}\| = \|\mathsf{R}\mathbf{x}\|. \tag{2.30}$$

Notons que les matrices de rotation sont en bijection avec l'ensemble des rotations.

2.7.2 Décomposition par angles élémentaires des matrices de rotation

La décomposition par angles élémentaires est connue dans le cas 3D sous le nom des angles d'Euler. Nous présentons les cas 2D et 3D en détails, puis le cas général.

Soit R_{2D} une matrice de rotation en dimension $p = 2$. Cette matrice a 1 degré de liberté et peut être paramétrée à l'aide d'un paramètre θ, l'angle de la rotation, comme :

$$\mathsf{R}_{2D}(\theta) = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}. \tag{2.31}$$

En dimension $p = 3$, la matrice de rotation R_{3D} a 3 degrés de liberté et peut être paramétrée par la composition de 3 rotations élémentaires, c'est à dire autour des trois axes x, y et z du repère monde, notées R_x, R_y et R_z. Il y a plusieurs possibilités pour composer ces 3 rotations élémentaires. Cette paramétrisation est appelée les *angles d'Euler*. Nous utilisons la forme suivante, où $\boldsymbol{\theta}$ est un vecteur de taille (3×1) encapsulant les angles d'Euler :

$$\mathsf{R}_{3D}(\boldsymbol{\theta}) = \mathsf{R}_x(\theta_1)\mathsf{R}_y(\theta_2)\mathsf{R}_z(\theta_3). \tag{2.32}$$

Cette paramétrisation est minimale, mais non-linéaire et admet des singularités car certaines rotations peuvent être représentées par plusieurs combinaisons d'angles d'Euler. Les rotations élémentaires sont définies par :

$$\mathsf{R}_x(\theta_1) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_1) & -\sin(\theta_1) \\ 0 & \sin(\theta_1) & \cos(\theta_1) \end{pmatrix} \tag{2.33}$$

$$\mathsf{R}_y(\theta_2) = \begin{pmatrix} \cos(\theta_2) & 0 & \sin(\theta_2) \\ 0 & 1 & 0 \\ -\sin(\theta_2) & 0 & \cos(\theta_2) \end{pmatrix} \tag{2.34}$$

$$\mathsf{R}_z(\theta_3) = \begin{pmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 \\ \sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.35}$$

Cette paramétrisation peut être étendue à une dimension p quelconque. Une matrice $\mathsf{R} \in SO(p)$ peut être décomposée en une composition de $p(p-1)/2$ rotations élémentaires, et paramétrée par les $p(p-1)/2$ angles de rotation. Chaque rotation élémentaire est construite, comme dans le cas 3D, en laissant invariant $p-2$ axes.

Il est important de noter que les paramétrisations à l'aide de rotations élémentaires sont non-singulières localement, c'est à dire pour de petits angles de rotation. Notons aussi que des angles de rotation nuls donnent des matrices identité : $\mathsf{R}_{2D}(0) = \mathrm{I}_{(2\times2)}$ et $\mathsf{R}_{3D}(\mathbf{0}_{(3\times1)}) = \mathrm{I}_{(3\times3)}$.

2.7.3 Estimation des matrices de rotation par mise à jour locale minimale

Le but de cette section est de donner un schéma d'estimation itératif d'une matrice de rotation, avec le minimum de paramètres, et sans singularité. Nous donnons un schéma d'estimation basé sur une mise à jour locale avec un nombre minimal de paramètres, basée sur la décomposition en angles élémentaires. Il existe d'autres possibilités de mise à jour, basées sur d'autres décompositions des matrices de rotation. Par exemple, Taylor et Kriegman [202] utilisent une paramétrisation basée sur l'exponentielle d'une matrice anti-symétrique.

L'idée est de paramétrer globalement les rotations par une matrice de rotation, qui est une paramétrisation redondante, puis de mettre à jour localement cette représentation par une paramétrisation minimale non-singulière localement. Nous utiliserons des schémas similaires dans nos algorithmes d'estimation géométrique.

Nous considérons ci-dessous les cas 2D et 3D. L'extension aux dimensions supérieures est triviale. Notons que ces schémas de mise à jour sont consistants, dans le sens où ils préservent les contraintes d'orthonormalité et de déterminant positif (2.29) sur les matrices de rotation manipulées, et sont complets dans le sens où toute matrice de rotation peut être atteinte à partir de toute initialisation. La formulation des schémas de mise à jour est placée dans le cadre de l'estimation itérative d'une matrice de rotation.

Cas 2D. Considérons le cas $p = 2$ et notons $\mathsf{R}_{2D,k}$ l'estimation de la matrice de rotation recherchée à l'itération k. Il est possible de faire une mise à jour locale de l'estimation en composant cette dernière avec une autre rotation :

$$\mathsf{R}_{2D,k+1} \quad = \quad \mathsf{R}_{2D,k}\mathsf{R}_{2D}(\theta_k), \tag{2.36}$$

où $\mathsf{R}_{2D}(\theta_k)$ est la paramétrisation d'une matrice de rotation 2D donnée par l'équation (2.31). Cette loi de mise à jour est basée sur 1 paramètre. Elle est donc minimale. La matrice $\mathsf{R}_{2D}(\theta_k)$ est une sorte de rotation de différence, initialisée à l'identité avant chaque itération, c'est à dire à $\theta_k = 0$. L'estimation consiste à calculer θ_k avec $\mathsf{R}_{2D,k}$ fixe, puis à mettre à jour cette matrice par la formule (2.36).

Le calcul du paramètre θ_k est souvent basé sur la minimisation d'une fonction de coût. Les algorithmes de type Newton évaluent ce θ_k en se basant sur la différenciation de la fonction de coût évaluée en l'estimation courante. Cette fonction de coût est exprimée en termes de $\mathsf{R}_{2D,k+1}$. L'utilisation de la loi de mise à jour (2.36) nécessite donc l'expression des variations de $\mathsf{R}_{2D,k+1}$ en termes de θ_k, évaluée en $\mathsf{R}_{2D,k}$, c'est à dire en $\theta_k = 0$. Nous cherchons donc à formuler :

$$\begin{aligned}
\left.\frac{\partial \mathsf{R}_{2D,k+1}}{\partial \theta_k}\right|_{\theta_k=0} &= \left.\frac{\partial \mathsf{R}_{2D,k}\mathsf{R}_{2D}(\theta_k)}{\partial \theta_k}\right|_{\theta_k=0} \\
&= \mathsf{R}_{2D,k} \left.\frac{\partial \mathsf{R}_{2D}(\theta_k)}{\partial \theta_k}\right|_{\theta_k=0} \\
&= \mathsf{R}_{2D,k} \left.\begin{pmatrix} -\sin(\theta_k) & -\cos(\theta_k) \\ \cos(\theta_k) & -\sin(\theta_k) \end{pmatrix}\right|_{\theta_k=0} \\
&= \mathsf{R}_{2D,k} \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix},
\end{aligned}$$

ce qui donne, si l'on note $\mathbf{r}_t$ les colonnes de $\mathsf{R}_{2D,k}$:

$$\left.\frac{\partial \mathsf{R}_{2D,k+1}}{\partial \theta_k}\right|_{\theta_k=0} \quad = \quad (\mathbf{r}_2 \quad -\mathbf{r}_1). \tag{2.37}$$

Le vecteur gradient correspondant est donné par la vectorisation de cette matrice. Notons que la simplicité de cette expression est due à l'évaluation en $\theta_k = 0$. Cette expression est identique à celle qui aurait été obtenue si une approximation au premier ordre de la rotation de mise à jour $\mathsf{R}_{2D}(\theta_k)$ avait été utilisée.

Cas 3D. Nous notons $\mathsf{R}_{3D,k}$ l'estimation de la matrice de rotation recherchée à l'itération k. La loi de mise à jour locale est donnée par :

$$\mathsf{R}_{3D,k+1} \quad = \quad \mathsf{R}_{3D,k}\mathsf{R}_{3D}(\boldsymbol{\theta}_k), \tag{2.38}$$

où $\mathsf{R}_{3D}(\boldsymbol{\theta}_k)$ est la paramétrisation par angles d'Euler définie par l'équation (2.32) et $\boldsymbol{\theta}_k^{\mathsf{T}} = (\theta_{k,1} \;\; \theta_{k,2} \;\; \theta_{k,3})$ est un vecteur-3.

Formulons la matrice Jacobienne de taille (9×3) de cette loi de mise à jour, donnant les variations de $\mathsf{R}_{3D,k+1}$ en fonction de $\boldsymbol{\theta}_k$ au premier ordre[3], autour de l'estimation $\mathsf{R}_{3D,k}$. Cette matrice est composée des vecteurs gradients de $\mathsf{R}_{3D,k+1}$ pour les 3 angles élémentaires :

$$\left.\frac{\partial \text{vect}(\mathsf{R}_{3D,k+1})}{\partial \boldsymbol{\theta}_k}\right|_{\boldsymbol{\theta}_k=0} \quad = \quad (\mathbf{a}_{k,1} \;\; \mathbf{a}_{k,2} \;\; \mathbf{a}_{k,3}),$$

où la notation suivante est employée :

$$\mathbf{a}_{k,l} \quad = \quad \text{vect}\left(\left.\frac{\partial \mathsf{R}_{3D,k}\mathsf{R}_{3D}(\boldsymbol{\theta})}{\partial \theta_{k,l}}\right|_{\boldsymbol{\theta}_k=\mathbf{0}}\right).$$

[3]La matrice Jacobienne est de taille (9×3) car elle donne les variations des coefficients de $\mathsf{R}_{3D,k+1}$ exprimée sous la forme $\text{vect}(\mathsf{R}_{3D,k+1})$.

Formulons les différents $\mathbf{a}_{k,l}$ en commençant par $\mathbf{a}_{k,1}$:

$$\mathbf{a}_{k,1} \quad = \quad \text{vect}\left(\left. \frac{\partial \mathsf{R}_{3D,k}\mathsf{R}_x(\theta_{k,1})\mathsf{R}_y(\theta_{k,2})\mathsf{R}_z(\theta_{k,3})}{\partial \theta_{k,1}} \right|_{\boldsymbol{\theta}_k = \mathbf{0}} \right).$$

La matrice $\mathsf{R}_{3D,k}$ ne dépend pas de $\boldsymbol{\theta}_k$: $\left. \frac{\partial \mathsf{R}_{3D,k}}{\partial \theta_{k,1}} \right|_{\boldsymbol{\theta}_k = 0} = \mathsf{R}_{3D,k}$. Les matrices $\mathsf{R}_y(\theta_{k,2})$ et $\mathsf{R}_z(\theta_{k,3})$ ne dépendent pas de $\theta_{k,1}$ et, évaluées en $\boldsymbol{\theta}_k = \mathbf{0}$, donnent l'identité, qui peut être éliminée de la formulation. Reste le facteur $\left. \frac{\partial \mathsf{R}_x(\theta_{k,1})}{\partial \theta_{k,1}} \right|_{\boldsymbol{\theta}_k = \mathbf{0}}$. Après utilisation de la formule (2.33), différenciation et évaluation en $\theta_{k,1} = 0$, on obtient :

$$\left. \frac{\partial \mathsf{R}_x(\theta_{k,1})}{\partial \theta_{k,1}} \right|_{\boldsymbol{\theta}_k = \mathbf{0}} \quad = \quad \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix}.$$

Nous remplaçons cette égalité dans le calcul ci-dessus, et en notant $\mathbf{r}_i$ les colonnes de $\mathsf{R}_{3D,k}$, nous obtenons :

$$\begin{aligned} \mathbf{a}_{k,1} &= \text{vect}\left(\mathsf{R}_{3D,k} \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix} \right) \\ &= \text{vect}\left(\begin{pmatrix} \mathbf{0} & \mathbf{r}_3 & -\mathbf{r}_2 \end{pmatrix} \right). \end{aligned} \tag{2.39}$$

Un raisonnement similaire conduit aux résultats suivants :

$$\begin{aligned} \mathbf{a}_{k,2} &= \text{vect}\left(\mathsf{R}_{3D,k} \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{pmatrix} \right) \\ &= \text{vect}\left(\begin{pmatrix} -\mathbf{r}_3 & \mathbf{0} & \mathbf{r}_1 \end{pmatrix} \right) \end{aligned} \tag{2.40}$$

$$\begin{aligned} \mathbf{a}_{k,3} &= \text{vect}\left(\mathsf{R}_{3D,k} \begin{pmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right) \\ &= \text{vect}\left(\begin{pmatrix} \mathbf{r}_2 & -\mathbf{r}_1 & \mathbf{0} \end{pmatrix} \right). \end{aligned} \tag{2.41}$$

2.8 Le modèle de caméra perspectif

Nous décrivons le modèle de caméra utilisé dans cette thèse. C'est le modèle de *caméra perspectif*. Nous n'utilisons pas le modèle de caméra affine et ses sous-classes, ni le modèle de caméra linéaire.

Le modèle de caméra perspectif, appelé aussi *modèle sténopé*, modélise une projection perspective, illustrée sur la figure 2.1. On parle du *centre de projection* et de la *rétine*. Soit $\boldsymbol{Q}$ un point de l'espace. Sa projection, le point $\hat{\boldsymbol{q}}$, est définie par l'intersection de la rétine avec le *rayon de projection*, formé par le centre de projection et le point $\boldsymbol{Q}$. Cette projection est modélisée par une transformation projective de $\mathbb{P}^3$ vers $\mathbb{P}^2$, représentée par une matrice P de taille (3×4). L'équation de projection est donnée par :

$$\hat{\mathbf{q}} \quad \sim \quad \mathsf{P}\mathbf{Q}.$$

Nous supposons que la *matrice de projection* P est toujours de rang plein, c'est à dire de rang 3. Elle admet donc un noyau droit de dimension 1, qui représente les coordonnées du centre de projection. C'est le seul point de l'espace ne pouvant être projeté par P. La matrice P étant définie à un facteur près, elle a $3 \times 4 - 1 = 11$ degrés de liberté. D'autres termes associés au modèle sténopé sont illustrés sur la figure 2.1 : l'*axe optique* passe par le centre de projection et est perpendiculaire à la rétine. Leur intersection définit le *point principal*. Le *plan focal* est parallèle à la rétine et contient le centre de projection.

Le lien entre la matrice de projection P et les grandeurs physiques caractérisant la caméra est établi en décomposant P en paramètres *intrinsèques* et en paramètres *extrinsèques*. Les paramètres intrinsèques sont parfois appelés les paramètres *internes*, le *calibrage* ou encore l'orientation interne de la caméra, et les paramètres

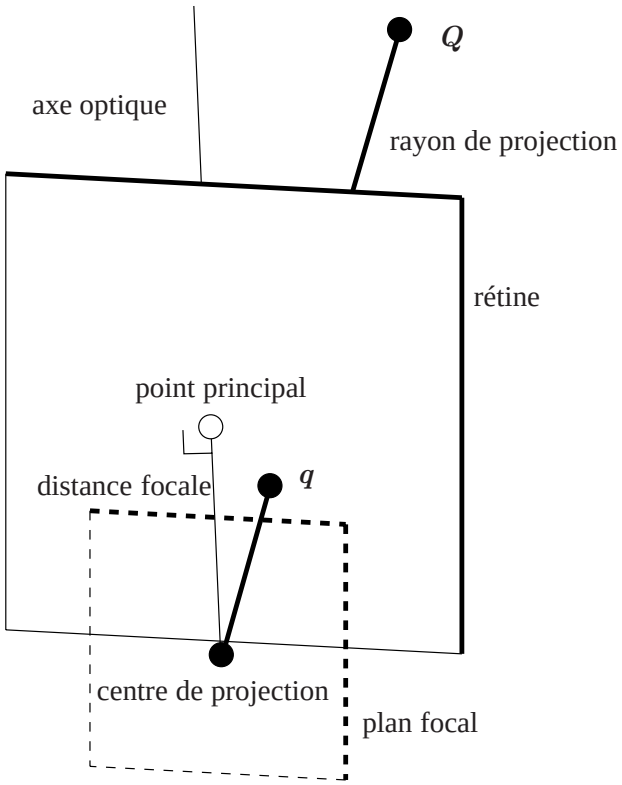


FIG. 2.1 – Le modèle de caméra perspectif ou sténopé.

extrinsèques la *pose* ou l'*orientation externe* de la caméra. Les paramètres intrinsèques sont des caractéristiques internes de la caméra, reliées par exemple à l'état du zoom. Ils comptent pour 5 des 11 degrés de liberté de la matrice P. Les paramètres extrinsèques définissent la position et l'orientation de la caméra, par une transformation rigide entre le repère monde, c'est à dire le repère global dans lequel sont exprimées les coordonnées des points 3D, et un repère caméra. Cette transformation Euclidienne encapsule les 6 autres degrés de liberté de la matrice P. L'origine du repère caméra est le centre de projection. L'axe x est parallèle à celui d'un repère 2D attaché à la rétine[4], et l'axe z coïncide avec l'axe optique.

Le processus de décomposition de P, détaillé par exemple dans [60, 93], conduit à l'équation suivante :

$$\mathsf{P} \sim \mathsf{K}(\mathsf{R} \quad \mathbf{t}).$$

Dans cette équation, K est une matrice (3×3) triangulaire supérieure modélisant les paramètres internes et R et $\mathbf{t}$ représentent respectivement la matrice de rotation et la translation modélisant la pose de la caméra.

Plus précisément, la matrice K s'écrit comme :

$$\mathsf{K} \sim \begin{pmatrix} \alpha f & \tau & u_0 \\ & f & v_0 \\ & & 1 \end{pmatrix}.$$

Elle réalise une transformation affine du repère rétine vers un repère image, et convertit des coordonnées exprimées avec une unité de mesure métrique en pixels. Dans cette formulation, f représente la *distance focale* ou tout simplement la *focale* de la caméra, c'est à dire la distance entre la rétine et le centre de projection, exprimée en pixels. Les coordonnées $(u_0 \;\; v_0 \;\; 1)^{\mathsf{T}}$ donnent la position du point principal (en pixels) dans le repère image. Finalement, α et τ modélisent respectivement le *rapport d'échelle*, c'est à dire le rapport entre les tailles

[4] Notons que l'axe y est parallèle à celui du repère 2D attaché à la rétine seulement si l'angle entre les axes du repère de la rétine est droit.

horizontale et verticale d'un pixel et l'angle entre les axes du repère de la rétine. Pour une caméra récente, les hypothèses $\alpha = 1$ et $\tau = 0$ n'altèrent pas la validité du modèle.

Les paramètres intrinsèques peuvent être extraits par décomposition QL de la matrice P normalisée telle que la norme de la dernière ligne soit égale à 1. Cette décomposition n'a de sens que si la caméra est calibrée, c'est à dire que si la matrice P est exprimée dans un repère de l'espace métrique. En anticipant sur la section suivante, on peut formuler cette condition en disant que l'ambiguïté sur la base de reconstruction, ou la liberté de jauge géométrique, doit être réduite à une transformation métrique. Le calibrage peut être estimé de manière statique, voir par exemple les travaux de Tsai [220] et de Lavest et al. [118], en observant un objet dont la structure 3D est connue a priori, ou par calibrage en ligne, voir par exemple les travaux de Maybank et Faugeras [137].

2.9 Stratification de la reconstruction et du calibrage, reconstructabilité

Nous examinons un critère de reconstructabilité, autrement dit, dans quels cas un scénario permet-il d'obtenir une solution unique de reconstruction. Une telle étude a des impacts sur la définition de la géométrie multi-vues. Notre raisonnement est valable pour le cas général : nous ne traitons pas les configurations singulières ou dégénérées et renvoyons aux travaux de Sturm [193, 194, 196] et Kahl et al. [106, 107, 108] pour un examen plus poussé de ces cas. Notons qu'il existe aussi des "surfaces critiques", c'est à dire dont la reconstruction est ambiguë, quel que soit le nombre de caméras et leur positionnement. Un travail récent sur ce sujet est [84]. Ces travaux montrent qu'il est très peu probable de rencontrer un cas de reconstruction singulier en pratique, sauf lorsqu'une phase de calibrage en ligne des caméras est effectuée.

La reconstructabilité d'un scénario est donnée par l'analyse du nombre de degrés de liberté $\mathcal{D}$ devant être déterminés et du nombre de contraintes $\mathcal{N}$ disponibles. La reconstructabilité est alors donnée par :

$\mathcal{N} - \mathcal{D} < 0$ La reconstruction est ambiguë ;
$\mathcal{N} - \mathcal{D} = 0$ La reconstruction est exacte mais la solution n'est pas forcement unique ;
$\mathcal{N} - \mathcal{D} > 0$ La reconstruction est sur-contrainte et calculable en minimisant une fonction de coût.

2.9.0.1 Etude du nombre de degrés de liberté

Une reconstruction est constituée de paramètres de structure et de mouvement[5], modélisés par des matrices et des vecteurs donnant leurs coordonnées exprimées dans une certaine base. Parmi ces paramètres, combien sont réellement libres et combien sont liés ? La réponse à cette question est donnée par l'étude du nombre de *degrés de liberté* du problème. C'est une question importante car la réponse conditionne la reconstructabilité. Elle permet aussi de déterminer si une paramétrisation est redondante ou minimale. Si cette étude est spécifique au problème considéré, la prise en compte de l'ambiguïté de la base de reconstruction est incontournable.

Par exemple, une caméra perspective a 11 degrés de liberté et un point 3D en a 3. Ils sont respectivement représentés par une matrice (3×4) et un vecteur-4 de coordonnées homogènes, ce qui revient bien à $3 \times 4 - 1 = 11$ et $4 - 1 = 3$ degrés de liberté. Notons que le nombre de degrés de liberté d'une droite 3D est 4.

Une reconstruction est modélisée par un ensemble de coordonnées exprimées dans une certaine base de l'espace 3D, le repère monde. Ce repère monde peut être changé arbitrairement, sans pour autant que la géométrie de la reconstruction soit changée. Toute reconstruction, quel que soit le niveau de calibrage et les primitives reconstruites, est soumise à l'ambiguïté du choix du repère monde. La dimension de cette ambiguïté, aussi appelée liberté de jauge géométrique, dépend du niveau de calibrage[6].

On peut alors calculer le nombre de degrés de liberté d'un système de n caméras et de m points par la formule $11n + 3m$. Il faut prendre en compte l'ambiguïté de la base de reconstruction. La dimension de cette ambiguïté est notée $\mathcal{D}_{\mathcal{A}}$, elle est donnée dans le tableau 2.1. Par exemple, $\mathcal{D}_{\mathcal{A}} = 15$ pour une reconstruction projective ou $\mathcal{D}_{\mathcal{A}} = 7$ pour une reconstruction métrique. Le nombre de degrés de liberté $\mathcal{D}$ d'une reconstruction de n caméras, m points et p droites est donné par la formule :

$$\mathcal{D} \quad = \quad 11n + 3m + 4p - \mathcal{D}_{\mathcal{A}}.$$

[5] Respectivement, les paramètres reliés à la surface de la scène observée et aux caméras, voir le chapitre 3 pour plus de détails.
[6] Plus de détails sont donnés sur les libertés de jauge en §3.4, chapitre 3.

2.9.0.2 Etude du nombre de contraintes

Les correspondances de primitives observées dans les images fournissent des contraintes sur la reconstruction. Chacun des t points mesurés et des u observations de droites fournit 2 contraintes sur la structure et le mouvement, d'où :

$$\mathcal{N} \quad = \quad 2(t+u).$$

En particulier, si tous les m points et les p droites sont visibles dans toutes les n vues, c'est à dire si il n'y a pas d'occultations, on obtient :

$$\mathcal{N} \quad = \quad 2n(m+p).$$

2.10 Géométrie d'un système de caméras non calibrées

Nous décrivons une modélisation de la géométrie multi-vues non calibrée. Par là, nous entendons la modélisation du positionnement relatif de n caméras dans l'espace projectif. Lorsque deux, trois ou quatre vues sont considérées, ce positionnement relatif est encapsulé par les *tenseurs d'appariement multi-vues*, la *matrice fondamentale*, le *tenseur trifocal* et le *tenseur quadrifocal*. Nous ne détaillons que le cas de deux vues, représenté par la matrice fondamentale modélisant la *géométrie épipolaire*. Nous présentons ensuite une modélisation du cas multi-vues basée sur la géométrie épipolaire et une collection de matrices de projection.

2.10.1 Le cas de deux vues : la géométrie épipolaire

Considérons deux caméras non calibrées et leurs matrices de projection $\mathsf{P} \sim (\,\bar{\mathsf{P}} \quad \mathbf{p})$ et $\mathsf{P}' \sim (\,\bar{\mathsf{P}}' \quad \mathbf{p}')$ exprimées dans une certaine base projective. Chacune de ces matrices (3×4) a 11 degrés de liberté. L'indétermination de la base projective retranche 15 degrés de liberté, ce qui donne $2 \times 11 - 15 = 7$ degrés de liberté à la géométrie des deux caméras. Ces 7 degrés de liberté sont encapsulés par la matrice fondamentale F entre les deux caméras. Cette dernière se déduit des matrices de projection, voir par exemple [128], comme :

$$\mathsf{F} \quad \sim \quad [\mathbf{p}' - \bar{\mathsf{P}}'\bar{\mathsf{P}}^{-1}\mathbf{p}]_\wedge \bar{\mathsf{P}}'\bar{\mathsf{P}}^{-1}.$$

La matrice fondamentale est invariante à la base de l'espace projectif où sont exprimées les caméras. Elle constitue une reconstruction implicite des deux caméras. Etant donnée une matrice fondamentale, le choix de matrices de projection "compatibles", appelées *réalisation* de la matrice fondamentale, nécessite de choisir une base de l'espace projectif. Le calcul d'une réalisation a donc une ambiguïté de dimension 15. Toutes les réalisations possibles d'une matrice fondamentale peuvent se déduire à partir d'une seule, par changement de base projective, c'est à dire multiplication des matrices de projection par une homographie 3D non singulière. La connaissance de la matrice fondamentale entre deux vues est parfois appelée *calibrage projectif* ou *calibrage faible* des caméras.

2.10.1.1 Interprétation géométrique

La matrice fondamentale constitue une reconstruction implicite des deux caméras et l'exprime par des contraintes sur la position des *correspondances de points*, c'est à dire les points image issus de la projection d'un même point 3D. Elle définit les projections des rayons de projection issus d'une caméra, dans l'autre image. Ce sont les droites épipolaires. Elles forment un faisceau dans chaque image. Les points d'incidence de ces deux faisceaux sont les épipôles, notés $\boldsymbol{e}$ et $\boldsymbol{e}'$, dont les coordonnées sont encapsulées par les noyaux gauche et droit de la matrice fondamentale :

$$\begin{aligned} \mathsf{F}\mathbf{e} &= \mathbf{0} \\ \mathsf{F}^\mathsf{T}\mathbf{e}' &= \mathbf{0}. \end{aligned}$$

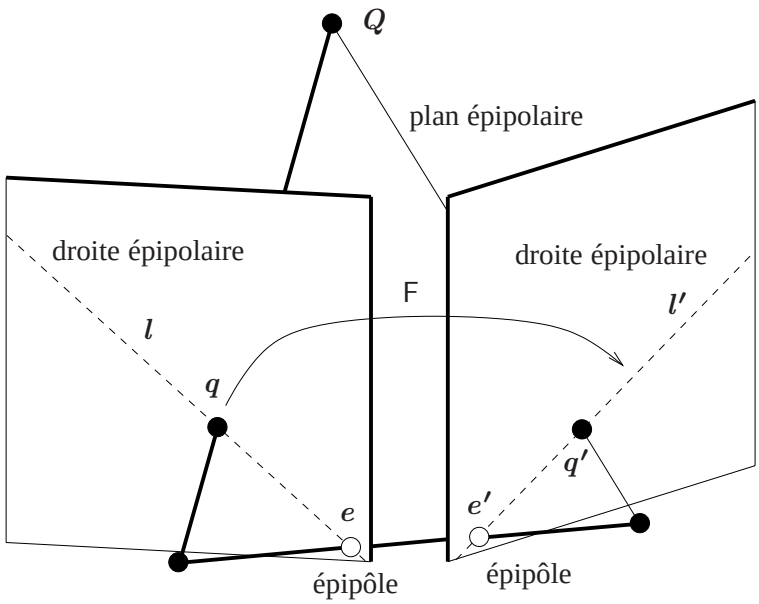


FIG. 2.2 – La géométrie épipolaire.

L'extraction des épipôles est étudiée en §2.10.1.4. Chaque faisceau de droites épipolaires est un espace projectif de dimension 1. La matrice fondamentale contient la correspondance entre ces faisceaux, une transformation homographique de $\mathbb{P}^1$. Elle associe à un point $\boldsymbol{q}$ de l'image de gauche, la droite épipolaire $\boldsymbol{l}$ correspondante dans l'image de droite, et inversement, à un point $\boldsymbol{q'}$ de l'image de droite, la droite épipolaire correspondante dans l'image de gauche :

$$\begin{aligned} \mathbf{l}' &\sim \mathsf{F}\mathbf{q} \\ \mathbf{l} &\sim \mathsf{F}^{\mathsf{T}}\mathbf{q}'. \end{aligned}$$

Les droites épipolaires peuvent aussi être formulées directement à partir des épipôles :

$$\begin{aligned} \mathbf{l} &\sim \mathbf{e} \wedge \mathbf{q} \\ \mathbf{l}' &\sim \mathbf{e}' \wedge \mathbf{q}'. \end{aligned}$$

A partir de la matrice fondamentale, on peut définir une contrainte très forte : la *contrainte épipolaire* existe pour tout couple de points image $\boldsymbol{q} \leftrightarrow \boldsymbol{q'}$ résultant de la projection d'un même point 3D :

$$\mathbf{q}'^{\mathsf{T}}\mathsf{F}\mathbf{q} = 0. \tag{2.42}$$

Cette contrainte exprime le fait que chaque point doit se trouver sur la droite épipolaire donnée par son correspondant dans l'autre image.

2.10.1.2 Réalisations canoniques

Réaliser une matrice fondamentale consiste à en extraire deux matrices de projection. Lors de cette étape, s'effectue le choix de la base de reconstruction, d'où les notions de base et de réalisation *canoniques*. Nous considérons les réalisations canoniques de la forme proposée par Hartley [90] :

$$\mathsf{P} \sim (\mathrm{I} \quad \mathbf{0}) \tag{2.43}$$

$$\mathsf{P}' \sim (\mathsf{H}_r \quad \gamma\mathbf{e}'), \tag{2.44}$$

où H_r est une homographie 2D appelée *homographie de référence* (voir ci-dessous), $\boldsymbol{e'}$ le deuxième épipôle et γ un facteur d'échelle entre H_r et $\mathbf{e}'$. On note, sans toutefois rentrer dans les détails, que ces réalisations

sont liées à la structure affine relative introduite par Shashua et Navab [179]. Il existe d'autres possibilités de réalisation canonique que nous n'abordons pas ici, voir en particulier les travaux de Faugeras [57] et de Heyden [96]. La famille des réalisations canoniques du type explicité ci-dessus est de dimension 4 : l'ambiguïté de base projective est de dimension 15, le choix de la matrice P retranche 11 degrés de liberté. Les différentes réalisations de cette forme tournent donc autour du choix de H_r et de γ. Si $\mathsf{P}' \sim (\ \mathsf{H}_r \quad \gamma\mathbf{e}')$ est valide, alors $\mathsf{P}'' \sim (\ \mathsf{H}_r + \mathbf{e}'\mathbf{a}^\mathsf{T} \quad \gamma'\mathbf{e}')$ l'est aussi, $\forall \mathbf{a} \in \mathbb{R}^3$ et $\forall \gamma' \in \mathbb{R}^*$. Le facteur γ n'a de sens que si H_r et $\mathbf{e}'$ sont normalisés. Luong et Viéville [130] et Hartley [90] proposent le choix suivant, que nous appelons la *réalisation canonique singulière* :

$$\check{\mathsf{P}}' \ \sim \ (\ \check{\mathsf{H}} \quad \mathbf{e}') \quad \text{avec} \quad \check{\mathsf{H}} \ = \ [\mathbf{e}']_\wedge \mathsf{F}. \tag{2.45}$$

Luong et Viéville appellent $\check{\mathsf{H}}$ la projection épipolaire. Ils fixent l'échelle relative entre $\check{\mathsf{H}}$ et $\mathbf{e}'$ en ramenant la norme de $\mathbf{e}'$ à 1. Ce choix est non-ambiguë car $\check{\mathsf{H}}$ est unique[7]. Des détails sont donnés en §4.2.1, chapitre 4. Le terme réalisation canonique singulière provient du fait que $\check{\mathsf{H}}$ est de rang 2. Ceci définit une caméra particulière, appelée caméra projective infinie [93, §5.3.6]. C'est une caméra dont le centre de projection se situe sur le plan à l'infini, mais dont le plan focal est différent du plan à l'infini (ce n'est donc pas une caméra affine).

Ceci n'est pas gênant pour la plupart des applications car la *base projective canonique singulière* ainsi définie est tout à fait valide. Il existe cependant des cas où une telle configuration est gênante. Par exemple, une caméra projective finie, c'est à dire dont la première sous-matrice (3×3) est de rang plein, est nécessaire pour la méthode de reconstruction de plans à partir de correspondances de points sur deux vues proposée en section 5.4.1.2, chapitre 5. Pour cette raison, nous proposons la *réalisation canonique non singulière*, formée via le choix d'une homographie de référence non singulière. Cette homographie est nommée *homographie de plan canonique non singulière*. Elle est définie par :

$$\mathcal{H} \ \sim \ [\mathbf{e}']_\wedge \mathsf{F} + \mathbf{e}'\mathbf{e}^\mathsf{T}. \tag{2.46}$$

Le terme $\mathbf{e}'\mathbf{e}^\mathsf{T}$ assure que $\mathcal{H}$ est de rang plein. On peut aisément vérifier que, comme requis pour toute homographie de plan, $\mathcal{H}$ met en correspondance les épipôles :

$$\mathcal{H}\mathbf{e} \ \sim \ [\mathbf{e}']_\wedge \mathsf{F}\mathbf{e} + \mathbf{e}'\mathbf{e}^\mathsf{T}\mathbf{e} \ = \ \mathbf{0} + \|\mathbf{e}\|^2\mathbf{e}' \ \sim \ \mathbf{e}'.$$

Nous fixons donc notre *réalisation canonique non singulière* par la deuxième matrice de projection suivante :

$$\mathcal{P}' \ \sim \ (\ \underbrace{\frac{[\mathbf{e}']_\wedge \mathsf{F} + \mathbf{e}'\mathbf{e}^\mathsf{T}}{\|\mathbf{e}\|^2}}_{\mathcal{H}} \quad \mathbf{e}'). \tag{2.47}$$

L'échelle relative entre $\mathcal{H}$ et $\mathbf{e}'$ est fixée car $\mathcal{H}\mathbf{e} = \mathbf{e}'$. Prouvons que $\mathcal{H}$ est de rang plein. L'équation (2.46) se reécrit sous la forme suivante :

$$\mathcal{H} \ \sim \ \underbrace{(\ [\mathbf{e}']_\wedge \quad \mathbf{e}')}_{\mathsf{M}_1} \underbrace{(\ \mathsf{F}^\mathsf{T} \quad \mathbf{e})^\mathsf{T}}_{\mathsf{M}_2}.$$

Chacune des deux matrices (3×4) M_1 et M_2 a une forme particulière. La sous-matrice principale est de rang 2, et la quatrième colonne est son noyau gauche. Il est facile de prouver que de telles matrices sont de rang 3, en utilisant par exemple MAPLE. Ceci ne nous renseigne pas sur le rang du produit $\mathsf{M}_1\mathsf{M}_2^\mathsf{T}$, et donc de la matrice $\mathcal{H}$. Nous devons montrer qu'il n'existe aucun vecteur-3 $\mathbf{q}$ tel que $\mathcal{H}\mathbf{q} = \mathbf{0}$, c'est à dire que M_2^T n'envoie aucun vecteur $\mathbf{q}$ sur le noyau de M_1. Pour ce faire, on note que le noyau de M_1 est de dimension 1, et engendré par un vecteur-4 $\mathbf{n}^\mathsf{T} \sim (\ \mathbf{e}'^\mathsf{T} \quad 0)$. Examinons la partie supérieure du produit $\mathsf{M}_2^\mathsf{T}\mathbf{q}$. Elle est donnée par le vecteur-3 $\mathsf{F}\mathbf{q}$. Donc, pour que $\mathbf{q}$ soit dans le noyau de $\mathcal{H}$, il faut que $\mathsf{F}\mathbf{q}$ soit dans le noyau de M_1, c'est à dire : $\mathsf{F}\mathbf{q} \sim \mathbf{e}'$, ce qui est impossible. En effet, $\forall \mathbf{q}$, $\mathbf{e}'^\mathsf{T}\mathsf{F}\mathbf{q} = 0$, car $\mathbf{e}'$ est le noyau gauche de F. Ceci implique que $\forall \mathbf{q}$, $\forall \alpha \in \mathbb{R}^*$, $\mathsf{F}\mathbf{q} \neq \alpha\mathbf{e}'$.

[7] Notons que le signe de l'échelle relative entre $\check{\mathsf{H}}$ et $\mathbf{e}'$ n'est pas fixé.

2.10.1.3 Paramétrisation des points 3D dans une base canonique

Lorsqu'une réalisation canonique avec $\mathsf{P} \sim (\ \mathrm{I} \quad \mathbf{0})$ est utilisée, Hartley et Zisserman [93, §A.4] proposent une paramétrisation minimale des points 3D. Cette paramétrisation est basée sur le fait que si $\mathbf{Q}^\mathsf{T} \sim (\ \bar{\mathbf{Q}}^\mathsf{T} \quad q)$ sont les coordonnées d'un point 3D $\boldsymbol{Q}$ et si $\hat{\mathbf{q}}$ sont les coordonnées de son image dans la première vue, alors on peut écrire $\bar{\mathbf{Q}} \sim \hat{\mathbf{q}}$. Par suite, si ce point est observé dans l'image, cela implique $\hat{q}_3 \neq 0$ et donc $Q_3 \neq 0$, et les coordonnées de $\boldsymbol{Q}$ peuvent s'écrire :

$$\mathbf{Q} \sim \begin{pmatrix} Q_1 \\ Q_2 \\ 1 \\ Q_4 \end{pmatrix}. \tag{2.48}$$

Un des avantages de cette paramétrisation est qu'elle est minimale et linéaire. Elle peut donc facilement être incorporée dans les étapes d'initialisation et d'ajustement de faisceaux d'un système de reconstruction. De plus, elle permet que l'erreur de reprojection dans la première image, c'est à dire le carré de la distance Euclidienne entre le point mesuré et le point prédit par le modèle 3D, soit donnée par le carré de la distance algébrique entre ces deux points.

2.10.1.4 Extraction des épipôles

Les épipôles sont donnés par les noyaux gauche et droit de la matrice fondamentale. On peut les en extraire en calculant une décomposition en valeurs singulières de la matrice fondamentale. Lorsque cette décomposition n'est pas utile à autre chose, on peut l'éviter en calculant les épipôles par les formules suivantes. Soient $\mathbf{c}_i$ les colonnes de F et $\mathbf{r}_r$ ses lignes :

$$\mathbf{e} \sim (\mathbf{r}_1 \wedge \mathbf{r}_2) + (\mathbf{r}_2 \wedge \mathbf{r}_3) + (\mathbf{r}_3 \wedge \mathbf{r}_1) \tag{2.49}$$

$$\mathbf{e}' \sim (\mathbf{c}_1 \wedge \mathbf{c}_2) + (\mathbf{c}_2 \wedge \mathbf{c}_3) + (\mathbf{c}_3 \wedge \mathbf{c}_1). \tag{2.50}$$

Ces formules découlent du fait que $\mathbf{e}$ est le noyau droit de F, il est donc orthogonal à toutes les lignes de F. On peut donc définir $\mathbf{e} \sim \mathbf{r}_r \wedge \mathbf{r}_{r'}$, $\forall r \neq r'$. La matrice fondamentale étant de rang deux, il existe des cas dégénérés, par exemple lorsque les lignes r et r' sont linéairement dépendantes, d'où la nécessité d'inclure toutes les combinaisons dans la formule (2.49). Un raisonnement similaire peut être tenu dans le cas de $\mathbf{e}'$ en utilisant les colonnes de F à la place des lignes. Ces expressions sont utilisées au chapitre 4, §4.2.1, dans le cadre de la formulation d'une paramétrisation minimale de la famille des homographies de plan.

2.10.2 Le cas multi-vues

Lorsque plus de deux vues sont considérées, une modélisation du placement relatif des caméras est la suivante. Deux caméras sont choisies. Nous les appelons les *caméras de référence*. Elles sont utilisées pour définir le repère projectif 3D monde et sont modélisées par une matrice fondamentale. Les autres caméras sont chacune modélisées par une matrice de projection, exprimée dans le repère monde. Si l'on compte le nombre de degrés de liberté de cette modélisation, on s'aperçoit qu'elle est minimale, c'est à dire que le nombre de paramètres libres est égal au nombre de degrés de liberté physiques. En effet, la matrice fondamentale a 7 degrés de liberté, et chacune des $n-2$ autres caméras en a 11, d'où un total de $7 + 11(n-2)$ degrés de liberté, correspondant aux $11n - 15$ degrés de liberté physiques d'un système de n caméras non calibrées.

2.11 Observation d'un plan

Nous étudions l'observation d'un plan par deux caméras, exprimées dans une base canonique, c'est à dire telle que $\mathsf{P} \sim (\ \mathrm{I} \quad \mathbf{0})$. Un plan induit une correspondance point-point entre les deux images, modélisée par une *homographie de plan*. Cette homographie de plan est représentée par une matrice (3×3). Lorsque la géométrie

3D des deux caméras, en d'autres termes la matrice fondamentale, est connue, on montre que les homographies de plan forment une famille de dimension 3, ce qui correspond au fait qu'un plan ait 3 degrés de liberté.

Nous examinons une paramétrisation minimale des plans, l'expression de l'homographie de plan induite en termes de cette paramétrisation et quelques propriétés des homographies de plans. Nous donnons ensuite des formules pour le calcul de l'intersection de deux et trois plans.

Les formules données dans cette section sont principalement utilisées au chapitre 5, dans le cadre de la reconstruction de plans.

Une paramétrisation minimale : l'équation réduite. Les plans sont habituellement représentés par des vecteurs-4 homogènes, donnant leurs coordonnées dans la base de reconstruction. Considérons un plan $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi}^\mathsf{T} \sim (\ \bar{\boldsymbol{\Pi}}^\mathsf{T} \quad \Pi)$ l'équation de ce plan. Dans une base de reconstruction canonique, c'est à dire telle que $\mathsf{P} \sim (\ \mathrm{I} \quad \mathbf{0})$, on peut facilement vérifier que la condition $\Pi = 0$ caractérise les plans contenant le centre de projection de la première caméra. Un tel plan se projette sur une droite dans la première image et n'est donc pas observable dans cette vue. Ceci correspond au fait qu'en espace Euclidien, Π donne la distance entre $\boldsymbol{\Pi}$ et le centre de projection de la première caméra[8]. Les plans contenant ce centre de projection sont exclus de la paramétrisation. On peut donc supposer $\Pi \neq 0$, et écrire l'équation du plan $\boldsymbol{\Pi}$ sous la forme :

$$\boldsymbol{\Pi}_{(4\times 1)} \sim \begin{pmatrix} \bar{\boldsymbol{\Pi}}_{(3\times 1)} \\ \\ 1 \end{pmatrix},$$

où $\bar{\boldsymbol{\Pi}}$ est un vecteur-3 non homogène appelé *équation réduite* du plan. Un tel vecteur-3 est une paramétrisation minimale des 3 degrés de liberté du plan.

Notons que lorsque plus de deux vues sont considérées, un plan peut contenir le centre de projection de la caméra de référence (dont la matrice de projection est $(\ \mathrm{I} \quad \mathbf{0})$) et être observé dans deux autres vues. L'équation réduite ne peut donc être utilisée pour paramétrer les plans que dans le cas de deux vues.

Homographie plane induite. Un plan d'équation réduite $\bar{\boldsymbol{\Pi}}$ induit l'homographie de plan suivante entre les deux vues :

$$\mathsf{H}_{\boldsymbol{\Pi}} \sim \mathsf{H}_r - \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T}, \tag{2.51}$$

où $\mathsf{P}' \sim (\ \mathsf{H}_r \quad \mathbf{e}')$ est la deuxième matrice de projection. Notons que cette équation caractérise le sous-espace de dimension 3 (engendré par $\bar{\boldsymbol{\Pi}}$) des homographies 2D qui sont des homographies de plan, c'est à dire pour lesquelles il existe un plan inducteur.

Notons que la paramétrisation (2.51) des homographies de plan est équivalente aux 6 équations suivantes[9] :

$$\mathsf{F}^\mathsf{T}\mathsf{H}_{\boldsymbol{\Pi}} + \mathsf{H}_{\boldsymbol{\Pi}}^\mathsf{T}\mathsf{F} = 0.$$

Intersection de deux plans. L'intersection de deux plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi}'$ est une droite $\boldsymbol{L}$, lorsque les deux plans ne sont pas identiques. Il existe beaucoup de représentations des droites 3D. En particulier, deux plans constituent une représentation. On peut aussi construire les coordonnées de Plücker de $\boldsymbol{L}$ directement à partir des coordonnées de $\boldsymbol{\Pi}$ et de $\boldsymbol{\Pi}'$, voir la section suivante. Pour éviter de choisir une représentation des droites 3D, nous cherchons à exprimer les coordonnées de la projection $\boldsymbol{l}$ de la droite $\boldsymbol{L}$ directement en fonction des coordonnées de plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi}'$. Lorsque la reconstruction est exprimée dans une base canonique avec $\mathsf{P} \sim (\ \mathrm{I} \quad \mathbf{0})$, les coordonnées de tout point 3D $\boldsymbol{Q}$ s'écrivent $\mathbf{Q}^\mathsf{T} \sim (\ \hat{\mathbf{q}} \quad Q)$ où $\hat{\boldsymbol{q}}$ est l'image de $\boldsymbol{Q}$ dans la première vue. La condition $\boldsymbol{Q} \in \boldsymbol{L}$ s'écrit algébriquement $\mathbf{Q}^\mathsf{T}\boldsymbol{\Pi} = \mathbf{Q}^\mathsf{T}\boldsymbol{\Pi}' = 0$, d'où :

$$\begin{cases} \hat{\mathbf{q}}^\mathsf{T}\bar{\boldsymbol{\Pi}} + Q = 0 \\ \hat{\mathbf{q}}^\mathsf{T}\bar{\boldsymbol{\Pi}}' + Q = 0, \end{cases}$$

[8] Si $\boldsymbol{\Pi}$ est normalisé tel que $\|\bar{\boldsymbol{\Pi}}\| = 1$.

[9] Le membre gauche de cette équation est une matrice symétrique de taille (3×3), ce qui explique pourquoi le nombre d'équations n'est pas 9 mais bien 6.

où $\bar{\mathbf{\Pi}}$ et $\bar{\mathbf{\Pi}}'$ sont les équations réduites des plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi}'$ respectivement. Ce système se transforme en :

$$\begin{aligned} \hat{\mathbf{q}}^\mathsf{T}\bar{\mathbf{\Pi}} &= \hat{\mathbf{q}}^\mathsf{T}\bar{\mathbf{\Pi}}' \\ \hat{\mathbf{q}}^\mathsf{T}(\underbrace{\bar{\mathbf{\Pi}} - \bar{\mathbf{\Pi}}'}_{\mathbf{l}}) &= 0, \end{aligned} \tag{2.52}$$

d'où l'on extrait l'équation de $\boldsymbol{l}$ recherchée. L'équation de $\boldsymbol{l}'$, la reprojection de $\boldsymbol{L}$ dans l'autre image, est obtenue trivialement par application de l'homographie duale induite par un des deux plans, $\mathsf{H}^\star_{\boldsymbol{\Pi}}$ ou $\mathsf{H}^\star_{\boldsymbol{\Pi}'}$.

Intersection de trois plans. Considérons un troisième plan $\boldsymbol{\Pi}''$. Soit $\boldsymbol{Q}$ le point d'intersection des plans $\boldsymbol{\Pi}$, $\boldsymbol{\Pi}'$ et $\boldsymbol{\Pi}''$, que l'on suppose unique. Nous donnons une expression de $\hat{\mathbf{q}}$ puis de $\mathbf{Q}$. Soient $\boldsymbol{l}$, $\boldsymbol{l}'$ et $\boldsymbol{l}''$ les droites projection de $\boldsymbol{\Pi} \cap \boldsymbol{\Pi}'$, $\boldsymbol{\Pi}' \cap \boldsymbol{\Pi}''$ et $\boldsymbol{\Pi}'' \cap \boldsymbol{\Pi}$ dans la première vue. Il est facile de montrer que ces trois droites s'intersectent en un point unique, $\hat{\boldsymbol{q}}$, l'image du point $\boldsymbol{Q}$. En effet, $\hat{\mathbf{q}}$ est le noyau droit de la matrice $\mathsf{A} \sim (\mathbf{l}\ \mathbf{l}'\ \mathbf{l}'')^\mathsf{T}$ dont le déterminant est nul. On peut le vérifier en écrivant A en termes des équations réduites des plans, d'après l'équation (2.52) :

$$\mathsf{A}^\mathsf{T} \sim \begin{pmatrix} \bar{\mathbf{\Pi}} - \bar{\mathbf{\Pi}}' & \bar{\mathbf{\Pi}}' - \bar{\mathbf{\Pi}}'' & \bar{\mathbf{\Pi}}'' - \bar{\mathbf{\Pi}} \end{pmatrix},$$

où il est trivial de voir que toute ligne est une combinaison linéaire des deux autres. L'expression de $\hat{\mathbf{q}}$ est obtenue en utilisant l'intersection de deux des trois droites $\boldsymbol{l}$, $\boldsymbol{l}'$ et $\boldsymbol{l}''$:

$$\hat{\mathbf{q}} \sim (\bar{\mathbf{\Pi}} \wedge \bar{\mathbf{\Pi}}') + (\bar{\mathbf{\Pi}}' \wedge \bar{\mathbf{\Pi}}'') + (\bar{\mathbf{\Pi}}'' \wedge \bar{\mathbf{\Pi}}). \tag{2.53}$$

Elle est identique quelle que soit la paire de droites choisie. On peut déterminer $\boldsymbol{Q}$ en intersectant le rayon de projection de $\hat{\boldsymbol{q}}$ avec un des trois plans, ou directement en utilisant un produit vectoriel généralisé. Chaque coefficient de $\mathbf{Q}$ est donné par le déterminant d'une matrice (3×3) de coefficients des équations de plan. Plus précisément, considérons les contraintes de coplanarité $\mathbf{\Pi}^\mathsf{T}\mathbf{Q} = \mathbf{\Pi}'^\mathsf{T}\mathbf{Q} = \mathbf{\Pi}''^\mathsf{T}\mathbf{Q} = 0$ et définissons la matrice M de taille (3×4) comme :

$$\mathsf{M}^\mathsf{T} \sim \begin{pmatrix} \mathbf{\Pi} & \mathbf{\Pi}' & \mathbf{\Pi}'' \end{pmatrix}.$$

Les coordonnées de $\boldsymbol{Q}$ sont données par le noyau de la matrice M, dont l'expression analytique est :

$$\mathbf{Q} \sim \begin{pmatrix} \Pi''(\bar{\mathbf{\Pi}} \wedge \bar{\mathbf{\Pi}}') + \Pi(\bar{\mathbf{\Pi}}' \wedge \bar{\mathbf{\Pi}}'') + \Pi'(\bar{\mathbf{\Pi}}'' \wedge \bar{\mathbf{\Pi}}) \\ -(\bar{\mathbf{\Pi}}, \bar{\mathbf{\Pi}}', \bar{\mathbf{\Pi}}'') \end{pmatrix},$$

où $(\bar{\mathbf{\Pi}}, \bar{\mathbf{\Pi}}', \bar{\mathbf{\Pi}}'') = \bar{\mathbf{\Pi}}^\mathsf{T}(\bar{\mathbf{\Pi}}' \wedge \bar{\mathbf{\Pi}}'')$ est le produit mixte, voir §2.1.

2.12 Les coordonnées de Plücker

Il n'existe pas de représentation "naturelle" des droites 3D et beaucoup de représentations ont été proposées. Plusieurs de ces représentations sont étudiées dans le livre de Hartley et Zisserman [93, §2.2.2]. Un état de l'art est présenté au chapitre 6 dans le cadre de la reconstruction de droites.

Les coordonnées de Plücker des droites 3D, introduites dans [159], ont l'avantage d'être complètes (toutes les droites de l'espace projectif $\mathbb{P}^3$ sont représentées) et de ne dépendre que de 6 paramètres, comparé par exemple à une représentation par deux points ou deux plans, nécessitant 8 paramètres. Un autre avantage majeur est qu'elles ne dépendent pas des points ou plans utilisés pour définir la droite. Förstner [69] propose une représentation unifiée (points, plans, droites, etc.) pour la géométrie projective algébrique, basée sur les coordonnées de Plücker pour représenter les droites 3D. Les coordonnées de Plücker sont parfois appelées coordonnées de Grassmann-Cayley car elles peuvent se construire à l'aide des opérateurs de cet algèbre. Nous renvoyons au livre de Faugeras et Luong [60] pour une présentation plus complète de cette approche. Notons

que le chapitre 6, §6.3.3 étudie la projection perspective d'une droite exprimée en coordonnées de Plücker, et que le chapitre 7 étudie l'application d'un mouvement 3D directement aux coordonnées de Plücker.

Les coordonnées de Plücker d'une droite $\boldsymbol{L}$ sont constituées d'un vecteur-6 de coordonnées homogènes noté $\mathbf{L}$. Nous considérons le partitionnement du vecteur $\mathbf{L}$ en deux vecteurs-3 $\mathbf{a}$ et $\mathbf{b}$, quelques fois appelés "partie supérieure" et "partie inférieure" :

$$\mathbf{L}^\mathsf{T} \;=\; (\underbrace{L_1 \;\; L_2 \;\; L_3}_{\mathbf{a}^\mathsf{T}} \;\; \underbrace{L_4 \;\; L_5 \;\; L_6}_{\mathbf{b}^\mathsf{T}}).$$

Il existe plusieurs définitions des coordonnées de Plücker. Quelle que soit la définition employée, le vecteur-6 est soumis à une contrainte de consistance bilinéaire, la *contrainte de Plücker*. Les coordonnées de Plücker ont donc un degré de liberté de jauge algébrique (le facteur d'échelle libre du vecteur-6) et une contrainte de consistance interne (la contrainte de Plücker).

Définition à partir de deux points. Soient $\mathbf{M} \sim (\;\bar{\mathbf{M}} \quad m)$ et $\mathbf{N} \sim (\;\bar{\mathbf{N}} \quad n)$ les coordonnées de deux points sur la droite $\boldsymbol{L}$, nous utilisons la définition suivante des coordonnées de Plücker :

$$\left\{ \begin{array}{rcl} \mathbf{a} & = & \bar{\mathbf{M}} \wedge \bar{\mathbf{N}} \\ \mathbf{b} & = & m\bar{\mathbf{N}} - n\bar{\mathbf{M}}. \end{array} \right. \tag{2.54}$$

Pour cette définition, la contrainte bilinéaire de Plücker est :

$$\mathcal{K}(\mathbf{L}) \;=\; \mathbf{a}^\mathsf{T}\mathbf{b} \;=\; 0. \tag{2.55}$$

Il est important de noter que les coordonnées de Plücker ne dépendent pas des points utilisés pour définir la droite.

Définition à partir de deux plans. Soient $\mathbf{\Pi}_1^\mathsf{T} \sim (\;\bar{\mathbf{\Pi}}_1^\mathsf{T} \quad \Pi_1)$ et $\mathbf{\Pi}_2^\mathsf{T} \sim (\;\bar{\mathbf{\Pi}}_2^\mathsf{T} \quad \Pi_2)$ les coordonnées de deux plans contenant la droite $\boldsymbol{L}$, la formulation suivante des coordonnées de Plücker est induite :

$$\left\{ \begin{array}{rcl} \mathbf{a} & = & \Pi_1\bar{\mathbf{\Pi}}_2 - \Pi_2\bar{\mathbf{\Pi}}_1 \\ \mathbf{b} & = & \bar{\mathbf{\Pi}}_1 \wedge \bar{\mathbf{\Pi}}_2. \end{array} \right. \tag{2.56}$$

Comme dans le cas de la définition par deux points, les coordonnées de Plücker définies par l'intersection de deux plans sont indépendantes des plans choisis dans le faisceau de plans autour de la droite $\boldsymbol{L}$.

La matrice de Plücker. La matrice de Plücker est une représentation des droites 3D étroitement liée aux coordonnées de Plücker. C'est une matrice (4×4) définie à un facteur près, anti-symétrique, et sur laquelle la contrainte de Plücker se traduit par une contrainte de rang 2. Notons L la matrice de Plücker représentant la droite $\boldsymbol{L}$. Cette matrice est reliée aux coordonnées de Plücker $\mathbf{L}$ par :

$$\mathsf{L} \;\sim\; \begin{pmatrix} [\mathbf{a}]_\wedge & \mathbf{b} \\ -\mathbf{b}^\mathsf{T} & 0 \end{pmatrix},$$

et aux coordonnées des points définissant la droite par :

$$\mathsf{L} \;\sim\; \mathbf{M}\mathbf{N}^\mathsf{T} - \mathbf{N}\mathbf{M}^\mathsf{T}.$$

Il est de même possible de définir L en fonction des coordonnées de deux plans contenant la droite $\boldsymbol{L}$.

Soit G une matrice de mouvement usuelle (agissant sur les points) générique (4×4), on montre que la matrice de Plücker représentant la droite transférée est donnée par :

$$\mathsf{L}' \;\sim\; \mathsf{G}\mathsf{L}\mathsf{G}^\mathsf{T}. \tag{2.57}$$

Cette expression est bilinéaire en les coefficients de la matrice de mouvement.

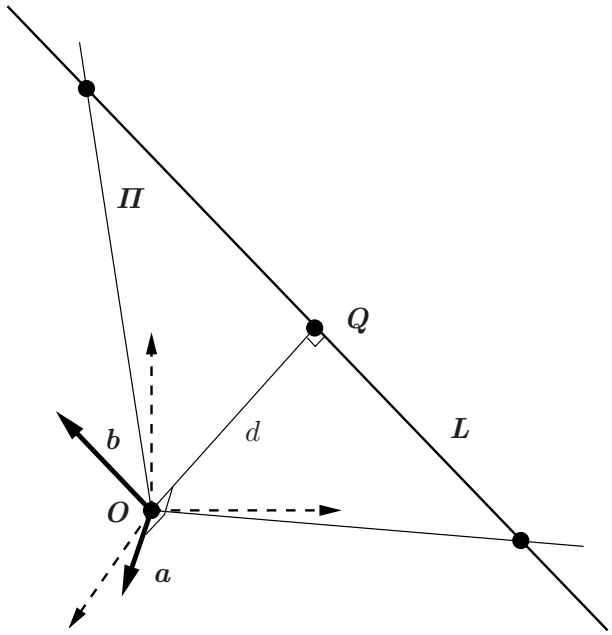


FIG. 2.3 – Interprétation géométrique des coordonnées de Plücker. Les axes du repère "monde" d'origine $\boldsymbol{O}$ sont en pointillés. Le plan défini par la droite $\boldsymbol{L}$ et $\boldsymbol{O}$ est noté $\boldsymbol{\Pi}$, $\boldsymbol{Q}$ est le point de $\boldsymbol{L}$ le plus proche de l'origine et d est la distance entre la droite et l'origine. Les vecteurs $\mathbf{a}$ et $\mathbf{b}$ forment les coordonnées de Plücker de $\boldsymbol{L}$: $\mathbf{L}^{\mathsf{T}} \sim (\ \mathbf{a}^{\mathsf{T}} \quad \mathbf{b}^{\mathsf{T}}\ )$ et s'interprètent respectivement comme la normale à $\boldsymbol{\Pi}$ et la direction de la droite. La distance d est proportionnelle au rapport suivant : $d = \|\mathbf{a}\|/\|\mathbf{b}\|$.

Interprétation géométrique des vecteurs $\mathbf{a}$ et $\mathbf{b}$. L'interprétation géométrique des coordonnées de Plücker est illustrée sur la figure 2.3. Considérons le plan défini par la droite et l'origine. Le vecteur-3 $\mathbf{a} = \bar{\mathbf{M}} \wedge \bar{\mathbf{N}}$ représente la direction de la normale à ce plan. Le vecteur-3 $\mathbf{b} = m\bar{\mathbf{N}} - n\bar{\mathbf{M}}$ est le point à l'infini de la droite, c'est à dire sa direction. On en déduit qu'un vecteur $\mathbf{b}$ nul est équivalent à une droite sur le plan à l'infini et qu'un vecteur $\mathbf{a}$ nul est équivalent à une droite contenant l'origine. Le rapport entre les normes des vecteurs $\mathbf{a}$ et $\mathbf{b}$ est égal à la distance d entre la droite et l'origine (en espaces Euclidien et métrique). Si ce rapport est infini, c'est à dire si $\mathbf{b} = \mathbf{0}$, on retrouve le fait qu'alors la droite est à l'infini. Si ce rapport est nul, c'est à dire si $\mathbf{a} = \mathbf{0}$, on en déduit que la droite contient l'origine.

Interprétation géométrique du vecteur $\mathbf{L}$. Le vecteur de coordonnées de Plücker $\mathbf{L}$ peut s'interpréter comme un point de l'espace projectif $\mathbb{P}^5$. La contrainte de Plücker étant bilinéaire, elle peut être écrite comme :

$$\mathcal{K}(\mathbf{L}) \quad = \quad \mathbf{L}^{\mathsf{T}} \mathsf{Q} \mathbf{L}.$$

Elle définit donc une quadrique dans $\mathbb{P}^5$. Cette quadrique est nommée "quadrique de Klein", "hypersurface de Plücker" ou encore "variété de Grassmann". Elle est représentée par la matrice (6×6) suivante :

$$\mathsf{Q} \quad \sim \quad \begin{pmatrix} 0_{(3\times3)} & \mathrm{I}_{(3\times3)} \\ \mathrm{I}_{(3\times3)} & 0_{(3\times3)} \end{pmatrix}.$$

CHAPITRE

3

CADRE GÉNÉRAL DE LA RECONSTRUCTION PAR AJUSTEMENT DE FAISCEAUX

Ce chapitre donne les principes de base et algorithmes standards pour la reconstruction par ajustement de faisceaux. Ces algorithmes sont non-linéaires et nécessitent une estimation initiale et une paramétrisation du problème. Après une description synthétique des méthodes d'initialisation, nous détaillons les fonctions de coût de moindres carrés non-linéaires mises en œuvre pour la reconstruction de points puis pour celle de droites. Nous examinons ensuite la phase d'optimisation non-linéaire. Nous détaillons le problème de la paramétrisation et définissons des critères permettant de juger de la qualité d'une paramétrisation. Nous décrivons les algorithmes de Gauss-Newton et de Levenberg-Marquardt et terminons le chapitre par le résumé d'un algorithme type d'ajustement de faisceaux.

Pour plus de détails sur l'ajustement de faisceaux, nous renvoyons aux travaux de Triggs et al. [219] (pour la communauté vision par ordinateur) et au recueil édité par Atkinson [6] (pour la communauté photogrammétrie).

3.1 Introduction

Nous considérons les caméras comme des capteurs. Les données transmises sont des images de la scène observée. De nombreux auteurs ont montré que ces images pouvaient être utilisées pour la détermination de la structure 3D. Ce processus est appelé reconstruction. Il implique, sauf cas particulier, la détermination conjointe de la position des caméras.

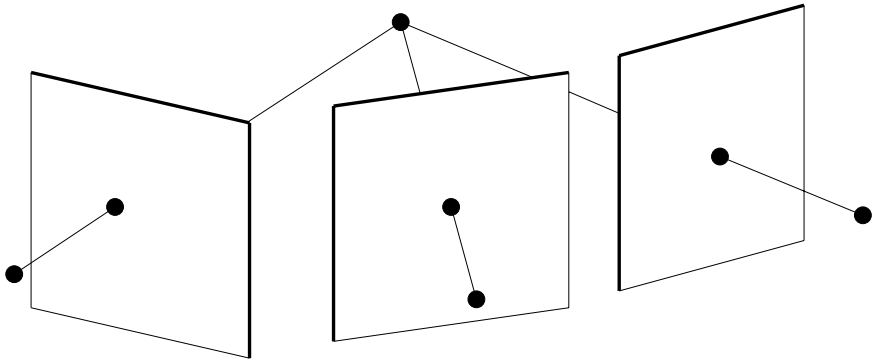


FIG. 3.1 – Reconstruction d'un point par intersection de rayons de projection.

L'idée intuitive sous-jacente à la reconstruction est que la position d'un point dans l'espace peut être déterminée par l'intersection de deux (ou plus) *rayons de projection*, voir figure 3.1. Ces rayons de projection doivent être issus de points image correspondant au point 3D reconstruit. De même, pour une droite, la primitive reconstruite peut être déterminée par l'intersection de *plans de projection*. La reconstruction nécessite donc une étape préalable d'extraction et de mise en correspondance de primitives entre les images. Nous supposons que cette étape a été effectué : nous disposons donc de *correspondances* de primitives image.

Nous nous concentrons sur la reconstruction de primitives éparses, telles les points et droites. Le problème de la reconstruction dense, où le but est de trouver la profondeur de chaque pixel, ne sera abordé qu'indirectement, au travers de la reconstruction de plans, au chapitre 5 et dans l'annexe A.

Les primitives et les caméras reconstruites sont respectivement nommées *structure* et *mouvement*. Les primitives image prédites à partir de la reconstruction sont nommées *reprojections*. La détermination conjointe de la structure et du mouvement est possible à partir de correspondances de primitives image. Les positions des rayons de projection correspondants sont contraintes. Par exemple, les rayons de projection correspondant à un même point 3D s'intersectent. Ceci fournit non seulement la structure, mais exerce de plus des contraintes sur le mouvement.

Les caméras sont des capteurs et les images qu'elles produisent ne sont pas un reflet exact de la scène observée. La luminance de chaque pixel est affectée par un *bruit de mesure*. La détection automatique ou la localisation manuelle des primitives image utilisées par nos algorithmes introduit de même un *bruit de détection* par rapport à la position idéale. La distribution du bruit total obtenu sur la position des primitives est ici supposée Gaussienne, centrée et distribuée de manière indépendante et identique ("identically independently distributed" ou iid en Anglais). C'est une hypothèse standard sur la distribution du bruit. Notons que Kanatani [110], Meer et al. [120, 136], et plus récemment, Okatani et Deguchi [157], étudient des modèles statistiques plus généraux.

A un système d'ajustement de faisceaux, est associée une *fonction de coût*, reflétant la qualité de l'estimation courante. Les systèmes d'ajustement de faisceaux permettent l'estimation des paramètres de la structure et du mouvement par minimisation de la fonction de coût associée. Cette fonction de coût est souvent choisie comme l'erreur de reprojection. Les minima globaux de l'erreur de reprojection correspondent à des solutions *optimales*. Le terme "optimal" est employé dans le sens du *maximum de vraisemblance*, sous les hypothèses de distribution statistique du bruit mentionnées ci-dessus, et en supposant que toutes les configurations sont équiprobables. Modéliser les caméras par une projection perspective induit la non-linéarité de l'erreur de reprojection pour les paramètres de la structure et du mouvement. L'utilisation de techniques d'optimisation non-linéaire est donc nécessaire, ce qui soulève les problèmes de l'*initialisation* et de la *paramétrisation* de la

structure et du mouvement.

Selon le niveau d'étalonnage ou de calibrage des caméras, la reconstruction obtenue est projective, affine, métrique ou Euclidienne. Les résultats donnés dans ce chapitre sont indépendants du calibrage des caméras. De même, ils sont valables pour des modèles de caméras différents du modèle perspectif (par exemple le modèle affine), incluant par exemple un modèle de distorsion optique des lentilles. Le propos sera illustré et les équations démontrées, sans perte de généralité, en utilisant le modèle de projection perspective, non calibré, avec des points. Nous renvoyons par exemple aux travaux récents de Lavest et al. [118] et aux travaux classiques en photogrammétrie [185] pour une étude des algorithmes d'ajustement de faisceaux calibrés incluant les paramètres internes des caméras (parfois appelés ajustement de faisceaux auto-calibrant en photogrammétrie).

Notons qu'il est courant, dans le domaine de la photogrammétrie, d'intégrer des contraintes géométriques aux systèmes d'ajustement de faisceaux. Ceci conduit à une formulation plus générale que celle présentée ci-dessous, voir par exemple le recueil edité par Atkinson [6]. Nous évoquerons l'incorporation de contraintes géometriques en §3.6.4, puis plus en détails au chapitre 5, dans le cas des contraintes de multi-coplanarité.

Organisation du chapitre. Nous donnons une brève description des méthodes d'initialisation en §3.2 puis examinons les fonctions de coût associées aux points et aux droites en §3.3. Les libertés de jauge sont mises en évidence et étudiées en §3.4. Nous détaillons le problème de la paramétrisation en §3.5 puis le principe des algorithmes de Gauss-Newton et de Levenberg-Marquardt en §3.6. Finalement, nous concluons et discutons ce chapitre en §3.7.

3.2 Initialisation

Le calcul d'une solution initiale au problème de la reconstruction est une étape difficile. Les fonctions de coût associées à l'ajustement de faisceaux sont non-linéaires et nécessitent d'être approximées, simplifiées. Des paramètres doivent être éliminés ou ajoutés, des grandeurs temporaires introduites, avant que la résolution de systèmes linéaires ne permette d'obtenir une solution sous-optimale de la reconstruction. Cette étape est critique dans le sens où le résultat va déterminer la convergence de l'ajustement de faisceaux vers le minimum local de la fonction de coût correspondant à la solution optimale.

Nous donnons un bref état de l'art des méthodes d'initialisation. Pour plus de détails sur ces méthodes, on renvoie par exemple à la thèse de Rother [169, §4.2]. Nous ne traitons pas les méthodes basées sur une petite distance entre caméras utilisant le flot optique ou des filtres récursifs tel le filtre de Kalman. Nous renvoyons à la thèse de Zucchelli [238] pour une revue de ces méthodes.

Nous classons les différentes méthodes en deux catégories : les méthodes directes et les méthodes séquentielles. Les méthodes directes retrouvent la structure et / ou les caméras en quelques étapes de calcul et sont invariantes à l'ordre des vues et des primitives. Les méthodes séquentielles initialisent la reconstruction à partir d'un petit nombre d'images et un sous-ensemble de primitives et reconstruisent itérativement les autres vues et primitives. Elles dépendent en général de l'ordre dans lequel les vues sont traitées.

Nous ne parlerons pas de l'utilisation des contraintes sur la structure, duales des tenseurs d'appariement multi-vues, au sens de la dualité de Carlsson [40], et renvoyons aux travaux de Hartley et Debunne [87] et de Schaffalitzky et al. [171].

3.2.1 Méthodes directes

3.2.1.1 Méthodes basées sur les tenseurs d'appariement

Les tenseurs d'appariement multi-vues – matrice fondamentale, tenseur trifocal et tenseur quadrifocal – encapsulent respectivement la géométrie de deux, trois et quatre caméras. Ils peuvent être estimés de manière linéaire à partir de correspondances de primitives image et permettent la reconstruction à partir d'ensembles de deux, trois ou quatre vues. Faugeras [57] et Hartley [88] introduisent la matrice fondamentale. Les tenseurs trifocal et quadrifocal ont alors été étudiés dans le cas des points et des droites, voir par exemple [62, 92, 98,

177, 181, 213]. Les cas d'autres primitives telles les coniques ont aussi été étudiés. Un résumé peut être trouvé dans les livres de Hartley et Zisserman [93] et de Faugeras et Luong [60].

La méthode standard pour l'estimation de la matrice fondamentale est l'algorithme des huit points normalisé donné par Hartley [86], présentée à l'origine par Longuet-Higgins [127] pour l'estimation de la matrice essentielle. Notons le travail récent de Fitzgibbon [65], qui propose un algorithme pour l'estimation simultanée de la matrice fondamentale et d'un terme de distorsion optique radiale.

Basé sur ces tenseurs, on peut extraire les caméras[1], voir par exemple [62, 97, 130]. Si les données comportent des appariements erronés, des techniques d'estimation robustes comme RANSAC [64] peuvent être utilisées, voir par exemple [209], ou son extension MLESAC proposée par Torr et Zisserman [206].

3.2.1.2 Méthodes de factorisation

L'idée de factoriser les caméras et la structure à partir d'une matrice de mesure a été introduite par Tomasi et Kanade [204]. Reid et Murray [165] montrent que la factorisation minimise l'erreur de reprojection. D'autres auteurs [103, 147] montrent comment l'incertitude sur la position des points peut être prise en compte.

Une version perspective de la factorisation est proposée par Sturm et Triggs [198]. Notons que dans ce cas, la matrice de mesure contient des profondeurs projectives inconnues. Ces facteurs ne sont pas arbitraires et doivent satisfaire certaines contraintes de consistance interne. Sturm et Triggs proposent des schémas de calcul séquentiel à partir des matrices fondamentales entre les différentes vues. Une solution alternative est de calculer ces profondeurs projectives itérativement, comme proposé indépendamment par Triggs [214] et Sparr [186]. De nombreux détails sont donnés dans le livre de Hartley et Zisserman [93]. Notons que dans le cas perspectif, contrairement au cas affine, la factorisation ne minimise pas l'erreur de reprojection, mais une erreur algébrique dépendant des points mesurés et reprojetés.

Le problème principal des algorithmes de factorisation est que la gestion des données manquantes est difficile. Tomasi et Kanade [204] suggèrent de choisir la plus grande sous-matrice de mesure pour calculer une reconstruction initiale. Trouver cette matrice est un problème NP-complet. Jacobs [104] propose une méthode consistant à insérer des combinaisons linéaires des colonnes existantes dans la matrice de mesure. Le problème de cette méthode est que la taille de la matrice de mesure augmente très rapidement avec la proportion de données manquantes. Martinec et Pajdla [134] combinent la factorisation perspective de Sturm et Triggs avec la méthode de Jacobs.

Quan et Kanade [164] montrent que la reconstruction de droites conjointement aux points est possible, mais requiert une étape préliminaire. Triggs [214] propose une méthode de factorisation de droites et de caméras, basée sur la génération de correspondances d'extrémités de droites entre les images à partir de tenseurs d'appariements et des correspondances de droites initiales. Martinec et Pajdla [135] proposent une méthode de factorisation de droites et de caméras basée sur les coordonnées de Plücker.

3.2.1.3 Méthodes par contraintes de fermeture

Les contraintes de fermeture ont été introduites par Triggs [216]. Ce sont des contraintes bilinéaires exercées par les tenseurs d'appariement multi-vues sur les matrices de projection. L'avantage par rapport aux algorithmes de factorisation est que la méthode traite facilement les données manquantes. Par contre, l'erreur minimisée est purement algébrique et la précision est en général moins bonne que celle d'un algorithme de factorisation. Triggs montre comment une phase préliminaire de calcul d'échelles projectives permet de déterminer toutes les matrices de projection comme la solution d'un système linéaire. Heyden et Kahl [99] montrent que dans le cas affine, les contraintes de fermeture sont linéaires pour les matrices de projection. La phase préliminaire de Triggs n'est donc pas nécessaire dans ce cas. La structure est ensuite reconstruite par triangulation. Les méthodes de Triggs et de Heyden et Kahl s'appliquent lorsque les caméras ne sont pas calibrées. Notons que Govindu [72] propose une approche similaire à celle de Triggs mais pour des caméras calibrées. Nous étudions dans l'article [73] l'application de ces méthodes à la reconstruction à partir de longues séquences en considérant l'aspect épars du problème.

[1] Comme par exemple les caméras formant la réalisation canonique, voir le chapitre 2, §2.10.1.2.

3.2.2 Méthodes séquentielles

Il existe deux grandes catégories de méthodes séquentielles. La première consiste à reconstruire à partir d'une sous-séquence à l'aide d'une méthode directe, puis à ajouter itérativement à la reconstruction les autres vues. Les matrices de projection sont calculées à l'aide de correspondances 3D-2D de primitives, puis des primitives supplémentaires sont éventuellement triangulées à partir de la nouvelle vue. Un système de reconstruction projective de cette catégorie, automatique et robuste, est présenté par Beardsley et al. [32]. Ils utilisent deux vues pour calculer la reconstruction initiale.

La deuxième catégorie de méthodes séquentielles est basée sur la fusion de reconstructions obtenues sur des sous-séquences à l'aide de méthodes directes. Fitzgibbon et Zisserman [66] proposent une approche de ce type. Leur méthode consiste à fusionner les reconstructions à partir de sous-séquences consécutives par estimation d'homographies 3D. A partir de tenseurs trifocaux, la structure et le mouvement sont calculés dans une base projective locale pour des triplets d'images. Les homographies 3D représentent les changements de base projective permettant d'aligner les bases locales. Le processus de fusion inclut une étape de raffinement des reconstructions partielles : lorsqu'un certain nombre de triplets sont alignés, un ajustement de faisceaux est lancé sur la sous-séquence, puis le processus de fusion est repris avec non plus le triplet d'images comme bloque de base, mais la sous-séquence fusionnée. Les reconstructions partielles sont fusionnées hiérarchiquement afin de distribuer l'erreur sur l'ensemble des images. L'ajustement de faisceaux peut être appliqué à tout moment pour réduire l'erreur. Notons qu'une des clefs de ces algorithmes est le calcul des homographies 3D. De nombreuses méthodes et critères existent dans le cas des points, mais très peu dans le cas des droites. Nous considérons ce problème au chapitre 8.

Ces méthodes sont appliquées avec succès à des séquences d'images. Nistér [155] recommande de ne pas traiter les images trop proches, ce qui pourrait déstabiliser la reconstruction. Ces idées sont utilisées et combinées avec des méthodes de calibrage en ligne, afin d'estimer une reconstruction métrique, voir par exemple les travaux de Pollefeys et al. [160].

L'application de ces méthodes n'est évidemment pas restreinte aux séquences d'images. Elles peuvent être appliquées à un ensemble non organisé d'images avec un grand nombre de données manquantes.

3.3 L'erreur de reprojection

Nous examinons les fonctions de coût associées à l'ajustement de faisceaux dans le cas des points puis des droites. Elles s'expriment sous la forme de sommes de carrés non-linéaires. Des fonctions de coût de même nature peuvent être formées pour la reconstruction d'autres primitives, telles les coniques. Le minimum global de ces fonctions de coût est obtenu pour des paramètres optimaux de la structure et du mouvement. Ces fonctions de coût ont une interprétation très intuitive. Elles sont minimisées par la structure et le mouvement donnant la "meilleure" prédiction des données, c'est à dire des primitives image. Elles reflètent la vraisemblance des paramètres estimés. Ces fonctions de coût dépendent donc des primitive images mesurées et de celles reprojetées. Ceci montre leur indépendance au modèle de projection, ainsi qu'au niveau de calibrage des caméras.

Les deux fonctions de coût décrites ci-dessous peuvent être utilisées indépendamment, pour la reconstruction de points ou de droites, ou être additionnées pour la reconstruction conjointe de points et de droites. Elles sont exprimées par la somme des carrés des distances entre les primitives mesurées et reprojetées. L'erreur de reprojection associée, exprimée en pixel, permet d'estimer la qualité d'une reconstruction et correspond à l'*erreur d'estimation*.

3.3.1 Le cas des points

Considérons un ensemble de m points 3D $\boldsymbol{Q}_j$ et un ensemble de n caméras P_i. La reprojection du point j dans l'image i est notée $\hat{\boldsymbol{q}}_{ij}$ et définie par :

$$\hat{\mathbf{q}}_{ij} \quad \sim \quad \mathsf{P}_i \mathbf{Q}_j .$$

La contribution de cette reprojection à la fonction de coût, illustrée sur la figure 3.2, est donnée par la différence entre le point reprojeté $\hat{\boldsymbol{q}}_{ij}$ et le point mesuré $\hat{\boldsymbol{q}}_{ij}$. Cette différence est mesurée par la distance Euclidienne

(2.13) entre ces deux points :

$$d_{PP}(\mathbf{q}_{ij}, \hat{\mathbf{q}}_{ij}) \quad = \quad \left\| \left(q_{ij,1} - \frac{\hat{q}_{ij,1}}{\hat{q}_{ij,3}} \quad q_{ij,2} - \frac{\hat{q}_{ij,2}}{\hat{q}_{ij,3}} \right) \right\|.$$

La fonction de coût est obtenue en additionnant les carrés des distances entre chacun des t points image mesurés et la reprojection correspondante. Définissons le *vecteur d'erreurs résiduelles* $\mathbf{r}$, de taille $(2t \times 1)$ par :

$$\mathbf{r}^{\mathsf{T}}_{(2t\times 1)} \quad = \quad \left(\quad \cdots \quad q_{ij,1} - \frac{\hat{q}_{ij,1}}{\hat{q}_{ij,3}} \quad q_{ij,2} - \frac{\hat{q}_{ij,2}}{\hat{q}_{ij,3}} \quad \cdots \quad \right).$$

On obtient alors l'expression suivante pour la fonction de coût :

$$\mathcal{Q}(\ldots, \hat{\mathbf{q}}_{ij}, \ldots) \quad = \quad \sum_{u=1}^{2t} r_u^2. \tag{3.1}$$

Selon les cas, on pourra omettre les arguments et noter simplement $\mathcal{Q}$. La minimisation de cette fonction de coût

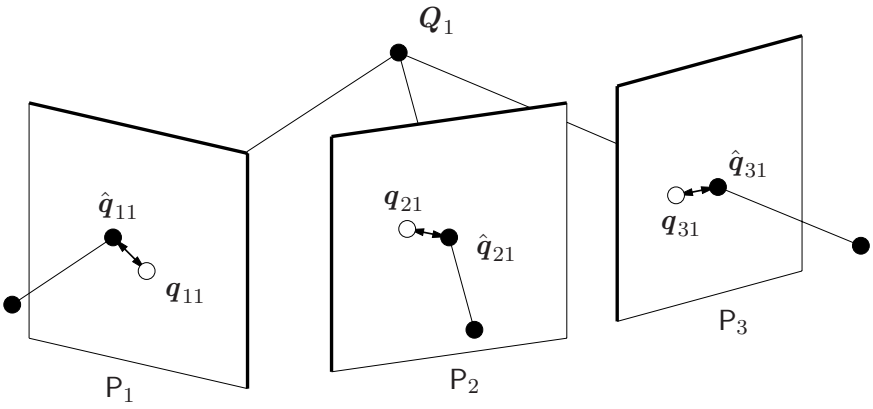


FIG. 3.2 – Reconstruction de point par ajustement de faisceaux : la fonction de coût est basée sur la distance entre les points image mesurés $\boldsymbol{q}_{ij}$ et les reprojections $\hat{\boldsymbol{q}}_{ij}$.

conduit à un résultat optimal, sous les hypothèses sur la distribution du bruit mentionnées plus haut. On peut voir la reconstruction de points comme le calcul de caméras et de points "corrigés" les plus proches possible des points mesurés et correspondant à un modèle 3D unique (intersection des rayons de projection).

3.3.2 Le cas des droites

Considérons un ensemble de p droites 3D $\boldsymbol{L}_k$. Contrairement aux points, il n'y a pas de représentation algébrique naturelle pour les droites 3D, voir le chapitre 6. Sans perte de généralité, nous utilisons les coordonnées de Plücker notées $\mathbf{L}_k$. La reprojection de la droite k dans l'image i est notée $\hat{\boldsymbol{l}}_{ik}$ et définie par :

$$\hat{\mathbf{l}}_{ik} \quad \sim \quad \tilde{\mathsf{P}}_i \mathbf{L}_k.$$

Dans cette équation, $\tilde{\mathsf{P}}_i$ désigne la matrice de projection (3×6) pour coordonnées de Plücker, définie par l'équation (6.3), associée à la matrice de projection P_i (voir le chapitre 6, §6.3.3). Le calcul de la contribution de cette reprojection à la fonction de coût nécessite de considérer quelles sont les données d'entrée, ou comment une droite image est définie.

Supposons sans perte de généralité que la droite image $\boldsymbol{l}_{ik}$ soit définie par deux points notés $\boldsymbol{x}_{ik}$ et $\boldsymbol{y}_{ik}$. Ce cas est rencontré en pratique lorsqu'un utilisateur détermine manuellement les droites image. Dans le cas où les droites sont détectées automatiquement, elles sont souvent définies comme le meilleur ajustement à un ensemble de plus de deux points. Ce cas est examiné plus loin. Notons la différence entre les *observations* et les *mesures* : une droite est observée, mais ce sont des points sur cette droite qui sont mesurés.

La contribution de l'observation de la droite k dans l'image i à la fonction de coût, illustrée sur la figure 3.3, est donnée par la somme des carrés des distances entre la droite reprojetée $\hat{\boldsymbol{l}}_{ik}$ et les points mesurés $\boldsymbol{x}_{ik}$ et $\boldsymbol{y}_{ik}$. La fonction de coût est obtenue en sommant ces contributions sur les s droites image observées. Définissons le vecteur d'erreurs résiduelles $\mathbf{r}$, de taille $(2s \times 1)$, par :

$$\mathbf{r}^{\mathsf{T}}_{(2s\times 1)} \;=\; \Big(\;\cdots\; d_{PH}(\mathbf{x}_{ik}, \hat{\mathbf{l}}_{ik}) \;\; d_{PH}(\mathbf{y}_{ik}, \hat{\mathbf{l}}_{ik}) \;\;\cdots\; \Big),$$

où $d_{PH}(\mathbf{x}_{ik}, \hat{\mathbf{l}}_{ik})$ est la distance entre le point $\boldsymbol{x}_{ik}$ et la droite $\hat{\boldsymbol{l}}_{ik}$ définie par l'équation (2.14). On obtient alors l'expression suivante pour la fonction de coût :

$$\mathcal{L}(\ldots, \hat{\mathbf{l}}_{ik}, \ldots) \;=\; \sum_{u=1}^{2s} r_u^2. \tag{3.2}$$

On note parfois $\mathcal{L}$ en fonction des paramètres $\mathcal{S}$ de la structure et $\mathcal{M}$ du mouvement : $\mathcal{L}(\mathcal{S}, \mathcal{M})$. Un résultat optimal pour la structure et le mouvement est obtenu par la minimisation de $\mathcal{L}$, sous les mêmes hypothèses de distribution statistique du bruit sur les points mesurés $\boldsymbol{a}_{ik}$ et $\boldsymbol{b}_{ik}$ que dans le cas de la reconstruction de points.

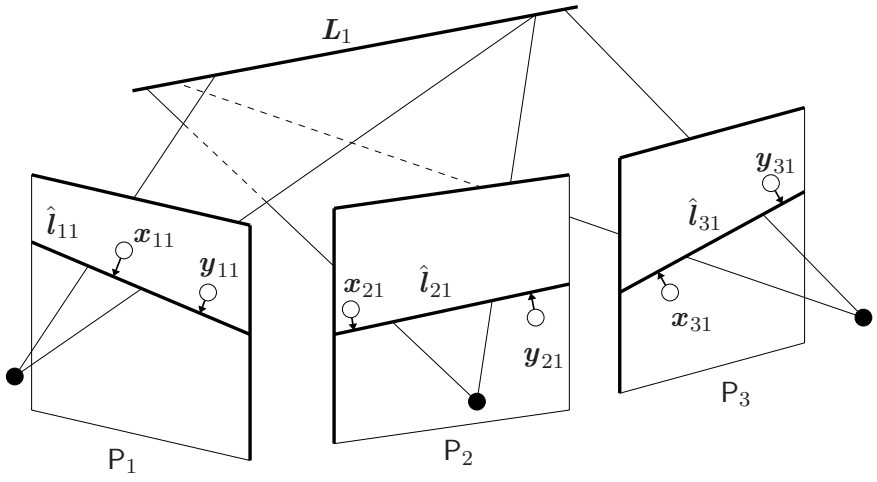


FIG. 3.3 – Reconstruction de droite par ajustement de faisceaux : la fonction de coût est basée sur la distance entre les points image mesurés $\boldsymbol{x}_{ik}$ et $\boldsymbol{y}_{ik}$ et les droites reprojetées $\hat{\boldsymbol{l}}_{ik}$.

Droites définies par plus de deux points. Lorsque les droites image sont définies par le meilleur ajustement à des ensembles de points, la fonction de coût peut être trivialement étendue pour prendre en compte tous ces points. Hartley [86] montre que sous certaines conditions de normalisation (permettant d'égaliser les distances Euclidiennes et algébriques), la somme des contributions de plusieurs points peut être exprimée par une constante non minimisable plus une somme pondérée de distances Euclidiennes avec deux points.

Pour tout groupe de points $\boldsymbol{c}_r$, il existe deux points $\boldsymbol{x}$ et $\boldsymbol{y}$ et deux facteurs scalaires non nuls α et γ tels que :

$$\sum_r d_{PH}^2(\mathbf{c}_r, \mathbf{l}) \;=\; \alpha^2 d_{PH}^2(\mathbf{x}, \mathbf{l}) + \gamma^2 d_{PH}^2(\mathbf{y}, \mathbf{l}) + \xi,$$

où ξ est une constante indépendante de $\mathbf{l}$. Ceci est valable si les coordonnées des points et de la droite sont normalisés tels que $d_{PH}^2(\mathbf{c}_r, \mathbf{l}) = (\mathbf{c}_r^{\mathsf{T}}\mathbf{l})^2$, c'est à dire si $\mathbf{c}_{r,3} = l_1^2 + l_2^2 = 1$. Dans ce cas, nous avons :

$$\textstyle\sum_r d_{PH}^2(\mathbf{c}_r, \mathbf{l}) \;=\; \sum_r \mathbf{l}^{\mathsf{T}}\mathbf{c}_r\mathbf{c}_r^{\mathsf{T}}\mathbf{l} \;=\; \mathbf{l}^{\mathsf{T}}\mathsf{C}\mathbf{l} \quad \text{avec} \quad \mathsf{C} \;=\; \sum_r(\mathbf{c}_r\mathbf{c}_r^{\mathsf{T}}).$$

Soit $\mathsf{J} = \begin{pmatrix} 1 & & \\ & 1 & \\ & & 0 \end{pmatrix}$, Hartley montre que C, matrice (3×3) symétrique de rang 3, peut s'écrire :

$$\mathsf{C}' \;=\; \mathsf{C} - \xi\mathsf{J},$$

où C' est une matrice symétrique de rang 2. La plus petite racine de l'équation $\det(\mathsf{C}) = 0$ donne ξ. Considérons la décomposition en valeurs singulières de C :

$$\mathsf{C}' \;=\; \mathsf{V}\,\mathrm{diag}(\sigma_1, \sigma_2, 0)\,\mathsf{V}^\mathsf{T} \;=\; \sigma_1 \mathbf{v}_1 \mathbf{v}_1^\mathsf{T} + \sigma_2 \mathbf{v}_2 \mathbf{v}_2{}^\mathsf{T},$$

qui donne :

$$\begin{aligned}\sum_r d^2(\mathbf{c}_r, \mathbf{l}) &= \mathbf{l}^\mathsf{T}(\sigma_1 \mathbf{v}_1 \mathbf{v}_1{}^\mathsf{T})\mathbf{l} + \mathbf{l}^\mathsf{T}(\sigma_2 \mathbf{v}_2 \mathbf{v}_2{}^\mathsf{T})\mathbf{l} + \xi \\ &= ((\sqrt{\sigma_1}\mathbf{v}_1)^\mathsf{T}\mathbf{l})^2 + ((\sqrt{\sigma_2}\mathbf{v}_2)^\mathsf{T}\mathbf{l})^2 + \xi \\ &= \sigma_1 v_{1,3}^2 d_{PH}^2(\mathbf{v}_1, \mathbf{l}) + \sigma_2 v_{2,3}^2 d_{PH}^2(\mathbf{v}_2, \mathbf{l}) + \xi,\end{aligned}$$

d'où :

$$\begin{aligned}\alpha^2 &= \sigma_1 v_{1,3}^2, \quad \mathbf{x} \sim \mathbf{v}_1 \\ \gamma^2 &= \sigma_2 v_{2,3}^2, \quad \mathbf{y} \sim \mathbf{v}_2.\end{aligned}$$

Nous utilisons l'erreur de reprojection pour les droites aux chapitres 6 et 8, pour la reconstruction de droites, puis l'alignement de reconstructions de droites. Dans ces chapitres, nous faisons l'hypothèse, sans perte de généralité, que l'erreur de reprojection induite par une droite dans une image, s'écrit à l'aide de la distance Euclidienne entre la droite reprojetée et deux points.

Observation. Le calcul d'une reconstruction de droites au maximum de vraisemblance requiert la reconstruction conjointe des points image définissant les droites. La fonction de coût est alors $\mathcal{Q}$, c'est à dire la somme des distances entre les points image et les points reprojetés, et l'optimisation est conduite sur les droites et les points reconstruits sur ces droites, ainsi que sur les caméras. La fonction de coût $\mathcal{L}$ donnée ci-dessus ne fait pas intervenir de points. Cependant, ces deux formulations sont équivalentes. En effet, les points reprojetés sont définis comme la projection orthogonale des points mesurés sur la droite reprojetée correspondante. Les deux fonctions de coût sont donc identiques. La reconstruction a posteriori des points les plus vraisemblants est triviale, après projection du points mesuré sur la droite reprojetée correspondante.

Autre fonction de coût. Lorsque l'on représente une droite 3D par deux points, il est tentant de baser la fonction de coût sur la distance entre les reprojections de ces deux points et une droite image ajustée aux points mesurés. Cette fonction de coût ne conduit pas à un résultat optimal car le bruit n'est pas modélisé directement où il apparait, c'est à dire sur les points image mesurés. Plus de détails sont donnés au chapitre 6.

3.3.3 L'erreur de reprojection et l'erreur d'estimation

L'erreur de reprojection est déduite des fonctions de coût associées à l'ajustement de faisceaux. Exprimée en pixel, elle permet d'évaluer la qualité d'une reconstruction. Elle correspond à l'*erreur d'estimation* au sens donné par [93, p.117], sous les hypothèses de distribution statistique du bruit faites plus haut. Dans le cas de la reconstruction exclusive de points ou de droites, l'erreur de reprojection est donnée respectivement par :

$$\mathcal{E}_\mathcal{Q} \;=\; \sqrt{\tfrac{\mathcal{Q}}{t}} \quad \text{et} \quad \mathcal{E}_\mathcal{L} \;=\; \sqrt{\tfrac{\mathcal{L}}{s}}.$$

Pour la reconstruction conjointe de points et de droites, elle est donnée par :

$$\mathcal{E}_{\mathcal{QL}} \;=\; \sqrt{\tfrac{\mathcal{Q}}{t} + \tfrac{\mathcal{L}}{s}} \;=\; \sqrt{\mathcal{E}_\mathcal{Q}^2 + \mathcal{E}_\mathcal{L}^2}.$$

3.4 Libertés de jauge

Les libertés de jauge géométriques, évoquées en §2.9, chapitre 2, sont une indétermination inhérente à toute reconstruction, qu'elle soit projective, affine, métrique ou Euclidienne. Ces ambiguïtés correspondent au choix

du repère monde où la reconstruction est exprimée. Les fonctions de coût données à la section précédente dépendent des primitives mesurées et de celles reprojetées. Leur valeur est invariante à la base de l'espace où la reconstruction est exprimée. D'autres libertés de jauge, les libertés de jauge algébriques, sont liées à la représentation algébrique de la reconstruction. Elles proviennent des échelles libres des représentations des points et des caméras en coordonnées homogènes. L'utilisation d'une distance Euclidienne dans les fonctions de coût entraîne leur indépendance à ces facteurs d'échelle. Dans le cas d'une reconstruction projective de m points observés par n caméras, la reconstruction a $15 + m + n$ degrés de liberté de jauge : pour toute transformation non singulière $\mathsf{G}_{(4\times4)}$, et $\forall\alpha_i \in \mathbb{R}^*, \forall\gamma_j \in \mathbb{R}^*$, les points de coordonnées $\gamma_j \mathsf{G}\mathbf{Q}_j$ et les caméras représentées par les matrices $\alpha_i \mathsf{P}_i \mathsf{G}^{-1}$ laissent invariantes les fonctions de coût. En effet, si l'on reprend l'équation de projection perspective (2.42), on obtient :

$$\hat{\mathbf{q}}_{ij} \quad \sim \quad \mathsf{P}_i\mathbf{Q}_j \quad \sim \quad (\alpha_i\mathsf{P}_i\mathsf{G}^{-1})(\gamma_j\mathsf{G}\mathbf{Q}_j).$$

Les libertés de jauge entraînent une ambiguïté dans la direction de recherche adoptée à chaque itération d'un système d'ajustement de faisceaux. Elles doivent être prises en compte explicitement dans la conception d'un tel système.

Il existe plusieurs stratégies de gestion des libertés de jauge, dont une revue est donnée par Triggs et al. [219, §9]. On peut classer les différentes méthodes en deux catégories : *globalement libres* (dérive libre de la jauge au cours de l'optimisation) et *globalement fixes* (jauge fixée globalement mais pas localement).

Les méthodes globalement libres sont basées sur la sélection d'une direction de recherche particulière en utilisant une méthode locale, c'est à dire dont le résultat est propre à chaque itération, sans tenir compte du chemin parcouru jusque là. La méthode de Gauss-Newton utilise une technique de pseudo-inverse [6], alors que la méthode de Levenberg-Marquardt utilise une technique de régularisation [121, 133, 174, 219]. Plus de détails sur ces techniques sont donnés dans la section suivante. Une autre possibilité est d'utiliser des contraintes de jauge locales. Par exemple, les contraintes internes photogrammétriques spécifient que les points ne doivent pas subir de translation, de rotation ou de changement d'échelle. Par contre, elles ne spécifient pas où se trouve le nuage de points dans l'espace [53, 219]. Ces contraintes possèdent d'autres propriétés : elles minimisent la norme du vecteur d'incrément et la variance des paramètres estimés.

Les méthodes globalement fixes utilisent une jauge choisie à l'avance. Certaines méthodes utilisent des éléments de référence, soit des points [6, 57, 145] (jauge centrée objet), soit des caméras [13, 89, 90]. Ces types de jauges sont appelés *jauges triviales*. Dans [6, p.40], l'auteur suggère que des points de contrôle ou des longueurs de référence soient utilisés pour fixer une base de reconstruction Euclidienne, alors que dans [57, 145], une méthode similaire basée sur 5 points de référence est utilisée pour fixer une base de reconstruction projective. Ces contraintes sont satisfaites en utilisant des multiplicateurs de Lagrange, ou par élimination [6, §2] et [57, 145]. Dans [90], les auteurs considèrent la reconstruction projective et fixent partiellement les jauges en utilisant une forme canonique pour une caméra de référence. Nous proposons dans [13] d'utiliser deux caméras de référence, ce qui fixe complètement la base projective. En espaces métrique ou Euclidien, une caméra (plus une échelle globale en espace métrique) suffit. Les facteurs d'échelle des coordonnées homogènes sont souvent fixés en normalisant ces dernières telles que leur norme soit 1 après chaque itération. Une telle jauge est souvent encapsulée dans la fonction de paramétrisation. Plus de détails sont donnés dans la section suivante.

Une autre possibilité de jauge globalement fixe est d'utiliser des contraintes de jauge globales [112, 140, 141], ce qui soulève le problème de l'inclusion de ces contraintes à l'optimisation. Dans [140], des observations artificielles sont ajoutées à la fonction de coût, avec un fort coefficient. Dans [112], la jauge est laissée libre durant l'itération, mais le résultat est projeté sur une jauge a posteriori, alors que dans [141], un sous-espace de paramètres projeté sur la jauge est calculé avant l'itération.

Les algorithmes d'optimisation non-linéaire utilisés ne sont pas invariant à la jauge[2]. Le choix d'une stratégie de gestion de la jauge influence donc le résultat des algorithmes. Il est difficile d'arrêter son choix sur une des méthodes existantes : peu de travaux ont effectué des comparaisons en termes de simplicité et d'efficacité.

[2]Les algorithmes proposés dans [112, 140, 141] sont dits invariants à la jauge, mais ne le sont qu'en projetant le résultat de chaque itération sur une jauge particulière prédéfinie, ce qui conditionne le résultat.

3.5 Paramétrisation

Nous posons le problème de la paramétrisation de la structure et du mouvement de manière générale. Nous abordons en détail le problème de la paramétrisation de la matrice fondamentale au chapitre 4, d'une structure plane par morceaux au chapitre 5 et des droites 3D au chapitre 6.

Une paramétrisation est une association entre les coefficients des représentations algébriques naturelles (par exemple les coordonnées homogènes de points 3D) et un jeu de paramètres. Une telle association est souvent appelée *carte* ("map" en Anglais). Le terme "carte" est utilisé en particulier par Ayache [8]. Une paramétrisation peut être basée sur plusieurs cartes, le choix de la meilleure carte dépendant de la valeur numérique des paramètres. Ayache appelle un ensemble de cartes un atlas. Nous utilisons simplement le terme *paramétrisation* pour exprimer un ensemble d'une ou plusieurs cartes.

Dans les sections suivantes, nous définissons la *fonction de paramétrisation* et la mise à jour des paramètres. Nous examinons les dépendances entre les libertés de jauge et la paramétrisation. Finalement, nous définissons un ensemble de critères permettant d'évaluer la qualité d'une paramétrisation. La figure 3.4 schématise le rôle de la paramétrisation pour l'optimisation itérative.

3.5.1 Motivations et fonction de paramétrisation

La non-linéarité des fonctions de coût pour les paramètres de la structure et du mouvement nécessite l'emploi de techniques d'optimisation non-linéaires. Soit $\mathcal{C}$ une fonction de coût, dépendant d'un vecteur $\mathbf{y}$ contenant les coefficients des entités recherchées. L'optimisation non-linéaire a pour but de résoudre des problèmes de la forme :

$$\min_{\mathbf{y}} \mathcal{C}(\mathbf{y}).$$

Certains problèmes ne peuvent être résolus, ou difficilement, en considérant directement les coefficients du vecteur $\mathbf{y}$. C'est le cas de certains des problèmes que nous abordons. Une paramétrisation du problème, modélisée par une fonction de paramétrisation $\mathcal{P}$, est alors nécessaire :

$$\mathbf{x} \quad = \quad \mathcal{P}(\mathbf{y}).$$

L'optimisation non-linéaire se reécrit en :

$$\min_{\mathbf{x}} \mathcal{C}(\mathcal{P}^{-1}(\mathbf{x})).$$

La fonction de paramétrisation définit un espace de recherche meilleur que celui constitué par les coefficients de la représentation naturelle. La taille du vecteur $\mathbf{x}$ est le *nombre de paramètres*. Si le nombre de paramètres est égal au nombre de degrés de liberté physiques du problème, la paramétrisation est *minimale*, sinon elle est *redondante*. De manière générale, la fonction de paramétrisation $\mathcal{P}$ est utilisée lors de l'initialisation. Son inverse, $\mathcal{P}^{-1}$, est ensuite utilisée à chaque itération, afin de calculer les valeurs de la fonction de coût pour l'estimation courante. Il est donc beaucoup plus important d'avoir une paramétrisation telle que $\mathcal{P}^{-1}$ plutôt que $\mathcal{P}$ consiste en un calcul simple et efficace.

3.5.2 Mise à jour

Soit k le compteur d'itérations d'un algorithme d'optimisation itératif. Le vecteur de paramètres $\mathbf{x}$ est mis à jour itérativement au cours de l'optimisation. La loi de mise à jour dépend de la paramétrisation employée. Elle prend souvent une forme additive :

$$\mathbf{x}_{k+1} \quad = \quad \mathbf{x}_k + \boldsymbol{\delta}_k. \tag{3.3}$$

Elle peut aussi être spécialisée aux différentes paramétrisations. Par exemple, dans le cas des matrices de rotation, les lois de mise à jour présentées en §2.7.3, chapitre 2, sont multiplicatives. Lorsque le nombre de paramètres de la loi de mise à jour correspond au nombre de degrés de liberté physiques du problème, on dit

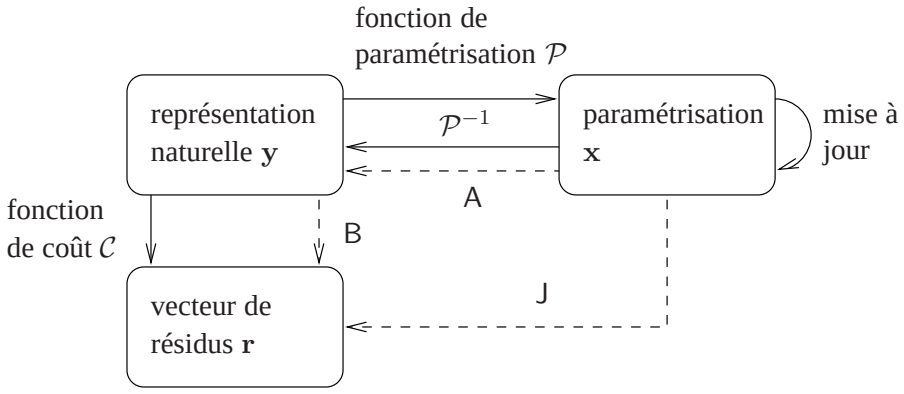


FIG. 3.4 – Illustration des notations et du rôle de la fonction de paramétrisation dans l'optimisation non-linéaire. Les traits en pointillés désignent les relations valables au premier ordre.

que la loi de mise à jour est *minimale*. C'est le cas des lois de mise à jour pour les matrices de rotation évoquées ci-dessus. La paramétrisation peut donc ne pas être minimale, tout en admettant une mise à jour locale minimale. C'est un critère important pour une paramétrisation.

Les algorithmes d'optimisation non-linéaire traditionnellement utilisés pour l'ajustement de faisceaux sont basés sur une prédiction de l'erreur nécessitant le calcul de la matrice Jacobienne J du vecteur d'erreurs résiduelles $\mathbf{r}$ associé à la fonction de coût, par rapport aux paramètres de mise à jour $\boldsymbol{\delta}$:

$$\mathsf{J} \;=\; \frac{\partial \mathbf{r}}{\partial \boldsymbol{\delta}}.$$

La prédiction de l'erreur étant effectuée autour de l'estimation courante, cette matrice est évaluée numériquement à l'estimation courante, c'est à dire que $\mathsf{J}|_{\boldsymbol{\delta}=\mathbf{0}} = \mathsf{J}|_{\mathbf{x}=\mathbf{x}_k}$ est utilisée. Notons que le rang de la matrice J n'est en général plein que si la jauge est fixée.

Il est pratique d'écrire cette matrice comme un produit de deux matrices A et B, dépendant respectivement uniquement de la fonction de coût et uniquement de la paramétrisation (et de la loi de mise à jour associée) :

$$\mathsf{J} \;=\; \mathsf{AB} \quad \text{avec} \quad \mathsf{A} \;=\; \frac{\partial \mathbf{r}}{\partial \mathbf{y}} \quad \text{et} \quad \mathsf{B} \;=\; \frac{\partial \mathbf{y}}{\partial \boldsymbol{\delta}}.$$

La matrice A est la matrice Jacobienne de la fonction de coût et la matrice B est celle de la loi de mise à jour associée à la paramétrisation employée. L'évaluation de cette matrice autour de l'estimation courante s'écrit $\mathsf{B}|_{\boldsymbol{\delta}=\mathbf{0}}$. Lorsque la mise à jour est additive, on peut écrire :

$$\mathsf{B} \;=\; \frac{\partial \mathcal{P}^{-1}}{\partial \mathbf{x}}.$$

Notons que dans ce cas, les formulations $\mathsf{B} = \frac{\partial \mathbf{y}}{\partial \mathbf{x}}\Big|_{\mathbf{x}=\mathbf{x}_k}$ et $\mathsf{B} = \frac{\partial \mathbf{y}}{\partial \boldsymbol{\delta}}\Big|_{\boldsymbol{\delta}=\mathbf{0}}$ sont équivalentes.

3.5.3 Influence des libertés de jauge

Les libertés de jauge se répercutent naturellement sur le vecteur de paramètres. Ceci implique que la matrice Jacobienne J a une déficience de rang égale au nombre de libertés de jauge, et induit des ambiguïtés dans le choix du vecteur $\boldsymbol{\delta}_k$, correspondant aux différentes jauges. La fonction de paramétrisation joue souvent un rôle lorsqu'une jauge triviale, basée sur l'élimination d'une ou plusieurs entités, est utilisée, voir l'exemple de la section 3.5.5. En effet, elle sert à éliminer la ou les entités nécessaires pour fixer (totalement ou partiellement) la jauge.

3.5.4 Evaluer la qualité d'une paramétrisation

Nous proposons une collection de critères permettant d'évaluer la qualité d'une paramétrisation. Il est important de noter que seules les expérimentations permettent vraiment de comparer différentes paramétrisations. Nous notons néanmoins au cours de cette thèse une forte adéquation entre les critères donnés ci-dessous et les comparaisons expérimentales. Ces critères sont :

- ▷ *minimalité* : la paramétrisation est-elle minimale ? Ou du moins admet-elle une loi de mise à jour minimale ? Si elle n'est pas minimale, alors c'est que des degrés de liberté de jauge n'ont pas été fixés et que des contraintes de jauge peuvent s'avérer nécessaires ;
- ▷ *linéarité* : plus la paramétrisation est linéaire, meilleur devrait être le résultat de chaque itération car le domaine de validité de l'approximation locale faite par l'algorithme d'optimisation est alors étendu ;
- ▷ *pas de liberté de jauge* : les degrés de liberté de jauge non fixés sont un désavantage ;
- ▷ *pas de contraintes internes* : les contraintes de consistance interne sont un désavantage si la loi de mise à jour n'est pas consistante, c'est à dire ne les garantit pas. Par exemple, les matrices de rotations sont soumises à des contraintes de consistance interne (l'orthonormalité des matrices, équation (2.29)), mais les lois de mise à jour (2.36) et (2.38) sont consistantes, dans le sens où elles préservent l'orthonormalité ;
- ▷ *complétude* : la paramétrisation peut-elle représenter l'ensemble des entités ? Par exemple, la paramétrisation des points 3D par un vecteur-3 $\bar{\mathbf{Q}}$ n'est pas complète car les points à l'infini ne sont pas représentables ;
- ▷ *différenciation analytique* : existe-t-il une expression analytique "simple" de la matrice Jacobienne décrivant les variations des coefficients de la représentation naturelle par rapport à celles des paramètres de mise à jour ? Par exemple, dans le cas des matrices de rotation, les équations (2.37), (2.39), (2.40) et (2.41) donnent les variations de la matrice de rotation en fonction des angles d'Euler ;
- ▷ *carte unique* : existe-t-il une unique carte entre les paramètres et les coefficients de la représentation naturelle ? Si plusieurs cartes sont nécessaires, un critère permettant de sélectionner la meilleure carte est requis. La présence de plusieurs cartes est un désavantage car l'implantation est en générale moins facile. Une matrice Jacobienne est nécessaire pour chaque carte. Il faut aussi gérer les "transitions" entre cartes, c'est à dire convertir les paramètres de la carte courante en paramètres équivalents pour la nouvelle carte.

Ces critères sont interdépendants. Ils font apparaître les points clefs de la qualité d'une paramétrisation, ce qui dépend aussi bien sûr de l'algorithme d'optimisation non-linéaire utilisé. Il est rare qu'une paramétrisation satisfasse tous ces critères.

3.5.5 Exemple de paramétrisation

Nous donnons un exemple de fonction de paramétrisation pour l'ajustement de faisceaux de caméras perspectives non calibrées et de points. Les fonctions de paramétrisation des caméras et des points sont notées respectivement $\mathcal{P}_{\mathsf{P}}$ et $\mathcal{P}_{\mathbf{Q}}$:

$$\mathbf{y}_{\mathsf{P}} \xrightarrow{\mathcal{P}_{\mathsf{P}}} \mathbf{x}_{\mathsf{P}} \quad \text{et} \quad \mathbf{y}_{\mathbf{Q}} \xrightarrow{\mathcal{P}_{\mathbf{Q}}} \mathbf{x}_{\mathbf{Q}},$$

où $\mathbf{y}_{\mathsf{P}}$ et $\mathbf{y}_{\mathbf{Q}}$ sont des vecteurs contenant les coefficients des caméras et des points respectivement. En reconstruction projective, il est courant de fixer 11 des 15 degrés de liberté de jauge géométrique en laissant invariante une caméra de référence. C'est la stratégie adoptée dans [93]. Par exemple, lorsque deux caméras P_1 et P_2 sont considérées et que la première caméra est choisie comme référence, $\mathbf{x}_{\mathsf{P}}$ est de taille 12 et défini par :

$$\mathbf{x}_{\mathsf{P}} \quad = \quad \text{vect}(\mathsf{P}_2),$$

où vect est l'opérateur de vectorisation d'une matrice. Cette paramétrisation est redondante.

De même pour les points, une paramétrisation redondante est souvent utilisée, correspondant à prendre pour $\mathcal{P}_{\mathbf{Q}}$ la fonction identité :

$$\mathbf{x}_{\mathbf{Q}}^{\mathsf{T}} \quad = \quad \mathbf{y}_{\mathbf{Q}}^{\mathsf{T}} \quad = \quad (\mathbf{Q}_1^{\mathsf{T}} \ \cdots \ \mathbf{Q}_m^{\mathsf{T}}).$$

Examinons les critères de la section 3.5.4 pour cette paramétrisation. Elle n'est pas minimale, mais elle est linéaire. Elle a $4 + (n-1) + m$ libertés de jauge, provenant d'une partie non fixée de la base projective, et des facteurs d'échelle des $n-1$ matrices de projection et des m points. Elle n'est pas soumise à des contraintes internes, elle est complète et la fonction de paramétrisation est basée sur une carte unique. La différenciation analytique est possible et la matrice Jacobienne est triviale.

3.6 Optimisation non-linéaire

L'ajustement de faisceaux est un problème d'optimisation de moindres carrés non-linéaires. Il existe des algorithmes globaux et locaux. Nous examinons deux algorithmes locaux souvent utilisés en vision par ordinateur et photogrammétrie, les algorithmes de Gauss-Newton et de Levenberg-Marquardt. Pour une présentation plus complète d'algorithmes d'optimisation non-linéaire continue, nous renvoyons aux livres de Seber et Wild [174], de Gill et al. [70] ou de Nocedal et Wright [156]. De manière plus générale, on peut identifier l'ajustement de faisceaux comme un problème de régression en distances orthogonales, dont l'étude fait l'objet des travaux de Boggs et al. [34]. Pour la communauté vision par ordinateur, on peut se référer au travail de Triggs et al. [219] et en photogrammétrie aux recueils édités par Atkinson [6] et par Slama [185].

L'idée de base est de mettre à jour une solution initiale $\mathbf{y}_0$ dont la paramétrisation est donnée par le vecteur $\mathbf{x}_0 = \mathcal{P}(\mathbf{y}_0)$. La loi de mise à jour correspondant à la paramétrisation est utilisée avec les paramètres de mise à jour $\boldsymbol{\delta}_k$. La différence entre les algorithmes réside dans la détermination des vecteurs $\boldsymbol{\delta}_k$. Les algorithmes de type Newton sont basés sur une approximation quadratique locale de la fonction de coût, formulée en termes des matrices Jacobienne $\mathsf{J} = \frac{\partial \mathbf{r}}{\partial \boldsymbol{\delta}}$ et Hessienne[3] $\frac{\partial^2 \mathbf{r}}{\partial^2 \boldsymbol{\delta}}$. L'évaluation de la matrice Hessienne est souvent coûteuse. L'approximation de Gauss-Newton permet d'écrire cette dernière en termes de la matrice Jacobienne : $\mathsf{N} = \mathsf{J}^\mathsf{T}\mathsf{J}$. Cette approximation est valide soit lorsque la fonction de coût est "proche" d'être linéaire, soit lorsque les résidus sont petits, ce qui est le cas dans la plupart des problèmes d'ajustement de faisceaux. Notons que, outre l'économie du calcul des dérivées secondes de la fonction de coût, l'approximation de Gauss-Newton confère à la matrice N une structure éparse, dans les cas de régressions en distances orthogonales, voir ci-après. Soit $\mathbf{g} = \mathsf{J}^\mathsf{T}\mathbf{r}$ le vecteur gradient de la fonction de coût. L'approximation quadratique s'écrit :

$$\mathcal{C}(\mathbf{x}+\boldsymbol{\delta}) \approx \mathcal{C}(\mathbf{x}) + \underbrace{\mathbf{g}^\mathsf{T}\boldsymbol{\delta} + \frac{1}{2}\boldsymbol{\delta}^\mathsf{T}\mathsf{N}\boldsymbol{\delta}}_{e}.$$

Le minimum de ce modèle quadratique[4] local simple est donné en posant $\frac{\partial e}{\partial \boldsymbol{\delta}} = 0$, ce qui donne les *équations normales* suivantes :

$$\mathsf{N}\boldsymbol{\delta} = -\mathbf{g}.$$

En présence de libertés de jauge, les équations normales ne sont pas de rang plein et plusieurs solutions sont admises. Les algorithmes de Gauss-Newton et de Levenberg-Marquardt proposent deux stratégies différentes de résolution.

Les sections suivantes examinent respectivement les algorithmes de Gauss-Newton et de Levenberg-Marquardt puis la structure des équations normales et différentes stratégies de résolution.

3.6.1 L'algorithme de Gauss-Newton

Dans sa version originale, l'algorithme de Gauss-Newton consiste à calculer la solution suivante pour $\boldsymbol{\delta}$:

$$\boldsymbol{\delta} = -\mathsf{N}^{-1}\mathbf{g}. \tag{3.4}$$

[3] La plupart des auteurs, par exemple Triggs et al. [219], utilisent une mise à jour additive, qui conduit à la formulation moins générale mais équivalente $\mathsf{J} = \frac{\partial \mathbf{r}}{\partial \mathbf{x}}$ et $\frac{\partial^2 \mathbf{r}}{\partial^2 \mathbf{x}}$ des matrices Jacobienne et Hessienne.

[4] Notons que ce modèle quadratique est basé sur deux approximations : un développement de Taylor autour des paramètres $\mathbf{x}$, puis l'approximation de Gauss-Newton $\mathsf{N} = \mathsf{J}^\mathsf{T}\mathsf{J}$ de la matrice Hessienne.

Cependant, en présence de libertés de jauge, la matrice N n'est pas de rang plein. Il existe alors une famille de solutions, dont la dimension correspond au nombre de degrés de liberté de jauge g. Cette famille de solutions peut être paramétrée par un vecteur $\mathbf{v}$ de taille $(g \times 1)$. Soit $\mathsf{N}^\sharp$ la pseudo-inverse de Moore-Penrose de la matrice N, et Λ une matrice dont les g colonnes forment une base du noyau de N, comme précisé en §2.6.3, chapitre 2, la famille de solutions s'écrit :

$$\boldsymbol{\delta} \quad = \quad -\mathsf{N}^\ddagger \mathbf{g} + \Lambda \mathbf{v}.$$

Une solution souvent utilisée, par exemple par Kanatani et Morris [112] est $\mathbf{v} = \mathbf{0}$, qui est la solution minimisant la norme du vecteur $\boldsymbol{\delta}$. Cette méthode laisse la jauge globalement libre.

3.6.2 L'algorithme de Levenberg-Marquardt

L'algorithme de Levenberg-Marquardt [121, 133] est une variante de Gauss-Newton consistant à effectuer une *régularisation* ou *augmentation* des équations normales. La matrice N est remplacée par une matrice régularisée $\underline{\mathsf{N}} = \mathsf{N} + \mathsf{W}(\lambda)$. La matrice $\mathsf{W}(\lambda)$, appelée matrice de régularisation, doit être définie positive pour garantir un rang plein à $\underline{\mathsf{N}}$. Les choix suivants ont été proposés pour $\mathsf{W}(\lambda)$:

- ▷ $\mathsf{W}(\lambda) = \lambda \mathrm{I}$. C'est le choix original de Levenberg et Marquardt [121, 133] repris par Triggs et al. [219]. Nous notons cette régularisation la méthode LM ;
- ▷ $\mathsf{W}(\lambda) = (1 + \lambda)\mathrm{diag}(\mathsf{N})$, c'est à dire multiplication des coefficients diagonaux de N par $(1 + \lambda)$. Ce choix est due à Seber [174]. Il est repris par Press et al. [163] et Hartley et Zisserman [93]. Nous notons cette régularisation la méthode SEBER.

Il est important de noter que ces choix pour la matrice $\mathsf{W}(\lambda)$ ne violent pas la structure éparse par bloques des équations normales (voir ci-dessous).

Le paramètre λ permet le contrôle de la régularisation. Sa valeur est ajustée de manière heuristique. Si l'itération réduit l'erreur, c'est à dire si $\mathcal{C}(\mathcal{P}^{-1}(\mathbf{x}_k + \boldsymbol{\delta}_k)) < \mathcal{C}(\mathcal{P}^{-1}(\mathbf{x}_k))$, alors la mise à jour est acceptée et la valeur de λ est divisée par une constante, souvent 10, sinon la mise à jour est rejetée et λ est multiplié par la constante.

Comme l'algorithme de Gauss-Newton, l'algorithme de Levenberg-Marquardt est de complexité cubique en le nombre de paramètres, par exemple, si n caméras et m points sont reconstruits, la complexité est en $O(n^3 m^3)$. Une résolution éparse des équations normales permet de réduire cette complexité, voir ci-dessous.

Une interprétation. Nous donnons une interprétation de cet algorithme, pour le cas où la matrice de régularisation est $\mathsf{W}(\lambda) = \lambda \mathrm{I}$. Lorsque λ est petit, $\underline{\mathsf{N}} = \mathsf{N} + \lambda \mathrm{I} \approx \mathsf{N}$, et on obtient une itération de type Gauss-Newton, donnée par l'équation (3.4). Lorsque λ est grand, $\underline{\mathsf{N}} = \mathsf{N} + \lambda \mathrm{I} \approx \mathrm{I}$, et on obtient alors une solution proche de $\boldsymbol{\delta} = -\mathbf{g}$, qui correspond à une itération du type descente de gradient.

Lien avec les régions de confiance. Les algorithmes de type Newton sont basés sur une approximation quadratique de la fonction de coût. L'idée des régions de confiance est de définir une région dans l'espace des paramètres, autour de l'estimation courante, dans laquelle l'approximation quadratique est supposée fiable. Le lien entre l'algorithme de Levenberg-Marquardt et les régions de confiance est établi par Moré [146]. La forme de la région de confiance est alors donnée par la norme définie par la matrice de régularisation $\mathsf{W}(\lambda)$. Le scalaire λ est lié au rayon de confiance. Notons qu'il existe beaucoup d'autres méthodes de contrôle d'itération ("step control" en Anglais), par exemple, les méthodes de "line search", pouvant éventuellement être combinées entre elles.

3.6.3 Structure et schéma de résolution éparse des équations normales

Les équations normales résolues à chaque itération ont en général une structure éparse par bloques. Cette structure est commune à tous les problèmes de régression en distances orthogonales. La prise en compte explicite de cette structure permet de résoudre les équations normales avec une complexité réduite. Cette solution exploite la structure éparse correspondant soit aux paramètres des points soit à ceux des caméras. La complexité

devient linéaire soit pour les points, soit pour les caméras, mais reste cubique pour l'autre, c'est à dire qu'elle est réduite de $O(n^3m^3)$ à $O(nm^3)$ ou à $O(n^3m)$.

Pour comprendre la résolution éparse des équations normales, il faut expliciter la structure des matrices Jacobienne et Hessienne de la fonction de coût. Si l'on organise le vecteur de paramètres avec les paramètres des caméras puis des points, et le vecteur de résidus en regroupant les résidus liés à chaque image, la matrice Jacobienne J a la forme dessinée sur la figure 3.5. L'approximation de Gauss-Newton de la matrice Hessienne est $\mathsf{N} = \mathsf{J}^\mathsf{T}\mathsf{J}$, sa structure est montrée sur la figure 3.5.

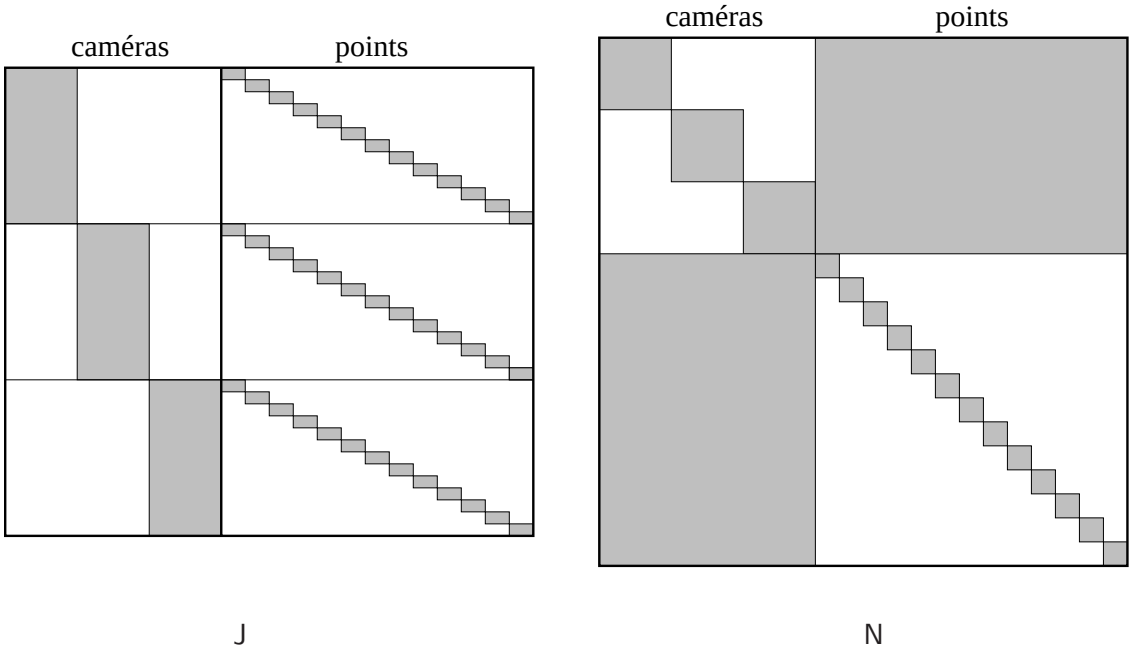


FIG. 3.5 – Structure des matrices Jacobienne J et Hessienne N (approximation de Gauss-Newton), déterminant les équations normales résolues à chaque itération de l'ajustement de faisceaux. Les bloques grisés contiennent les coefficients du systèmes, alors que les parties blanches ne contiennent que des zéros. Cet exemple est pour 3 caméras et 13 points visibles dans toutes les vues.

Si l'on choisit de résoudre de manière éparse par rapport aux points, la première étape consiste à résoudre de manière dense pour les caméras en éliminant les points, puis à résoudre de manière éparse pour les points. Nous renvoyons aux travaux spécialisés sur ce sujet pour plus de détails [34, 89, 185, 219].

Notons que des solutions de résolution éparse plus générales peuvent être mises en œuvre pour la résolution des équations normales, par exemple le gradient conjugué, voir par exemple [156], ou la factorisation LU éparse, voir par exemple [52].

Il existe par ailleurs des problèmes classiques en vision par ordinateur conduisant à des structures éparses différentes pour les équations normales. Par exemple, lors de la reconstruction métrique de caméras et de points, il est courant d'estimer des paramètres communs aux caméras. On voit alors apparaître une structure découpée en trois parties : les paramètres externes des caméras avec les paramètres internes spécifiques à chaque caméra (souvent la longueur focale), les paramètres internes communs (souvent le point principal) et les paramètres des points.

3.6.4 Reconstruction avec contraintes géométriques

On peut imaginer inclure des contraintes géométriques à la reconstruction. Au chapitre 5, nous abordons le cas des contraintes de coplanarité entre les points reconstruits. Les méthodes d'ajustement de faisceaux avec des contraintes ont été particulièrement étudiées dans la communauté des photogrammètres, voir par exemple le recueil édité par Atkinson [6]. Les méthodes proposées sont, entres autres, l'utilisation de multiplicateurs

de Lagrange, la linéarisation des contraintes et l'élimination de variables de manière analytique ou numérique. Nous renvoyons au chapitre 5 pour plus de détail sur ces méthodes.

3.7 Conclusions

Nous avons donné un cadre général des systèmes de reconstruction par ajustement de faisceaux. L'étape du calcul d'une solution initiale a été passée en revue, puis les fonctions de coût associées à la reconstruction de points et de droites ont été examinées en détail. Ces fonctions de coût sont liées à l'erreur de reprojection, qui est une mesure importante de qualité pour la reconstruction obtenue. Les problèmes induits par les libertés de jauge, inhérentes au problème comme le repère de reconstruction, ou dépendantes de la représentation comme les facteurs d'échelle des coordonnées homogènes, ont été examinés. L'étape de paramétrisation a été étudiée en détail, et des critères permettant d'évaluer la qualité d'une paramétrisation ont été proposés. Finalement, des méthodes d'optimisation de moindres carrés non-linéaires – les algorithmes de Gauss-Newton et de Levenberg-Marquardt – ont été passées en revue.

CHAPITRE

4

GÉOMÉTRIE DE DEUX CAMÉRAS NON CALIBRÉES ET ESTIMATION NON-LINÉAIRE DE LA MATRICE FONDAMENTALE

La géométrie de deux caméras non calibrées est modélisée par la matrice fondamentale. Son estimation joue un rôle clef en vision 3D par ordinateur. Elle est souvent effectuée à partir de correspondances de points entre deux images. Un critère de minimisation ayant un sens géométrique induit une fonction de coût non-linéaire.

Nous nous concentrons sur la compréhension du rôle joué par la matrice fondamentale dans la définition et la paramétrisation du positionnement relatif de deux caméras non calibrées.

Ce chapitre est divisé en trois parties. Dans la première partie, nous étudions différentes factorisations de la matrice fondamentale en épipôles et transformation épipolaire 1D. Nous proposons une représentation 2D de la transformation épipolaire, la *transformation épipolaire étendue*. Dans la deuxième partie, nous donnons un état de l'art détaillé des paramétrisations de la matrice fondamentale. Le formalisme précédemment introduit permet une construction simple de la plupart de ces paramétrisations. Il n'existe pas de paramétrisation minimale, globale et non-singulière de la matrice fondamentale. Les paramétrisations sont donc soit redondantes, soit basées sur plusieurs cartes. Dans la troisième partie, nous proposons la *représentation orthonormale* de la matrice fondamentale. Cette représentation est complète et permet l'estimation de la matrice fondamentale avec le nombre minimal de 7 paramètres.

Nos algorithmes sont finalement validés et comparés aux méthodes existantes sur des données simulées et réelles.

Les résultats concernant la transformation épipolaire étendue ont été obtenus en collaboration avec Peter Sturm et Radu Horaud et ont été publiés dans [28, 29]. Les travaux concernant la représentation orthonormale de la matrice fondamentale ont été effectués en collaboration avec Peter Sturm et ont été publiés dans [13, 22, 27].

4.1 Introduction

La matrice fondamentale joue un rôle clef en vision 3D par ordinateur. Elle représente une reconstruction implicite de deux caméras. Ces dernières peuvent en être extraites et utilisées pour reconstruire la structure 3D par triangulation de correspondances d'indices visuels. La matrice fondamentale est de taille (3×3), définie à un facteur près et de rang 2. Elle a donc 7 degrés de liberté. Son estimation est une étape préliminaire à beaucoup d'algorithmes de reconstruction et a donc une grande importance. La majeure partie des algorithmes d'estimation est basée sur des correspondances de points. L'algorithme des huit points proposé par Longuet-Higgins [126] et amélioré par Hartley [86] fournit une solution, minimisant un critère algébrique, souvent utilisée pour initialiser un algorithme non-linéaire, minimisant un critère géométrique. Triggs [218] classe les algorithmes non-linéaires en deux catégories : l'approche réduite, impliquant seulement la matrice fondamentale, et l'approche directe, impliquant en plus la structure 3D. On peut apparenter l'approche directe à un algorithme d'ajustement de faisceaux, car elle implique une reconstruction projective complète de la scène. L'optimisation non-linéaire pose le problème de la paramétrisation de la matrice fondamentale.

La géométrie de deux vues peut être représentée entre autres par la géométrie épipolaire ou par deux matrices de projection. La différence principale entre ces deux représentations est la suivante. La matrice fondamentale encapsule le *positionnement relatif* des deux caméras, et en constitue une reconstruction implicite. Les deux matrices de projection fixent le *positionnement absolu* des caméras, c'est à dire qu'elles dépendent du positionnement relatif des caméras et du choix de la base projective de reconstruction. Etant donnée une matrice fondamentale, on peut retrouver des matrices de projection en fixant la base de reconstruction. Tout couple de caméras ainsi obtenu est une *réalisation* de la matrice fondamentale.

Nous nous concentrons sur le problème de l'estimation non-linéaire et plus particulièrement le problème de la paramétrisation, qui revient à exhiber une ou plusieurs associations entre un jeu de paramètres et les coefficients de la matrice fondamentale. Une telle association est appelée "carte". Nous renvoyons au chapitre 3, §3.5, pour plus de détails.

Paramétrisations et méthodes existantes. La paramétrisation peut être basée entre autres sur les matrices de projection ou sur la matrice fondamentale. Un premier réflexe est d'associer ces deux possibilités avec les approches directe et réduite respectivement[1], ce qui présente des aspects pratiques indéniables, mais n'a pas de caractère obligatoire.

Baser la paramétrisation directement en fonction des matrices de projection conduit souvent à des paramétrisations redondantes car la base de reconstruction est souvent totalement ou partiellement incluse dans la paramétrisation. Par exemple, Hartley [90] propose de paramétrer la deuxième matrice de projection. Le nombre de paramètres à estimer est alors de 12. Parmi ces paramètres, 7 encapsulent le positionnement relatif des deux caméras et 4 fixent une partie de la base projective de reconstruction.

Baser la paramétrisation sur la matrice fondamentale conduit souvent à des paramétrisations minimales. Cette approche soulève plusieurs difficultés. La contrainte de rang 2 à laquelle est soumise la matrice fondamentale est non-linéaire. De plus, la matrice fondamentale est homogène et définie à un facteur près. L'ensemble des matrices fondamentales constitue une variété de dimension 7 dans $\mathbb{R}^9$. Cette variété est topologiquement non-triviale, ce qui signifie qu'on ne peut pas trouver 7 paramètres exprimant toutes les matrices fondamentales de manière simple et globale [217]. En conséquence, les cartes de la matrice fondamentale sont intrinsèquement locales, c'est à dire réduites à une sous-variété. Luong et Faugeras [128] proposent une paramétrisation avec 8 paramètres et une carte basée sur l'hypothèse que les deux épipôles ne sont pas à l'infini. Représenter de façon complète cette variété requiert un ensemble de cartes. C'est la stratégie adoptée par Zhang [233] et Csurka et al. [49]. Zhang propose une paramétrisation avec 8 paramètres et 9 cartes et Csurka et al. proposent une paramétrisation minimale (7 paramètres) avec 36 cartes.

L'approche de Zhang et Loop [236] permet d'éviter l'utilisation de cartes en se ramenant à la sous-variété paramétrée par Luong et Faugeras : ils proposent de ramener artificiellement les caméras dans une position où

[1] L'approche directe, équivalente à un ajustement de faisceaux, nécessite une forme explicite de la reconstruction des caméras, donnée directement par les matrices de projection, alors que la fonction de coût associée à l'approche réduite s'exprime directement en termes de la matrice fondamentale.

les épipôles sont finis, par une transformation projective des points image.

Torr et Zisserman [211, 206] proposent de représenter la matrice fondamentale par 7 correspondances de points. A chaque étape de l'optimisation non-linéaire, les coordonnées de ces points sont mises à jour et la matrice fondamentale correspondante est estimée.

Contributions. Ce chapitre contient deux contributions principales dont la lecture est indépendante.

La première contribution est une étude de la matrice fondamentale et de sa factorisation en épipôles et transformation épipolaire 1D. Nous introduisons la *transformation épipolaire étendue*, extension de la transformation épipolaire au 2D. Par rapport à la factorisation classique "épipôle + homographie", cette dernière permet d'établir un niveau supplémentaire de factorisation de la matrice fondamentale : "épipôle + épipôle + transformation épipolaire étendue". Ce formalisme permet de former simplement les paramétrisations minimales de la matrice fondamentale proposées par ailleurs [49, 128, 233]. Nous donnons un état de l'art détaillé de ces paramétrisations.

Notre deuxième contribution est une solution permettant d'optimiser la matrice fondamentale avec le nombre minimal de 7 paramètres, sans avoir recours à plusieurs cartes ni à un changement artificiel des positions des caméras. L'idée est de parcourir la variété des matrices fondamentales avec une paramétrisation basée sur la *représentation orthonormale* de la matrice fondamentale que nous proposons. Cette représentation a plusieurs avantages. Elle est générale, dans le sens où toutes les matrices fondamentales sont représentées de manière unifiée, et la matrice Jacobienne associée a une expression analytique simple, de même que la réalisation canonique.

Organisation du chapitre. Nous examinons la factorisation de la matrice fondamentale en épipôles et transformation épipolaire en §4.2. Nous y proposons la *transformation épipolaire étendue*. En §4.3, nous montrons comment elle permet une formulation simple des paramétrisations existantes au travers d'un état de l'art détaillé des paramétrisations et méthodes pour l'optimisation non-linéaire de la matrice fondamentale. Notre *représentation orthonormale* de la matrice fondamentale est donnée en §4.4, ainsi que son application à l'estimation non-linéaire de la matrice fondamentale pour différents critères (approches directe et réduite). Le chapitre se termine avec des résultats expérimentaux comparant notre approche à d'autres méthodes puis nos conclusions et perspectives en §§4.5 et 4.6 respectivement.

4.2 Factorisation de la matrice fondamentale

La matrice fondamentale définit un faisceau de droites épipolaires dans chaque image, ainsi qu'une correspondance homographique entre ces droites épipolaires. Elle encapsule le point d'incidence de chacun des faisceaux épipolaires, ce sont les *épipôles*, ainsi qu'une homographie de $\mathbb{P}^1$ à 3 degrés de liberté, nommée *transformation épipolaire*. Ces entités peuvent être extraites de la matrice fondamentale et inversement, cette dernière peut être exprimée en fonction de ces entités : c'est la *factorisation* de la matrice fondamentale.

Considérons la formulation suivante de la matrice fondamentale : $\mathsf{F} \sim [\mathbf{e}']_\wedge \mathsf{H}_{\boldsymbol{\Pi}}$, où $\boldsymbol{e}'$ est le deuxième épipôle et $\mathsf{H}_{\boldsymbol{\Pi}}$ une homographie de plan. On remarque que seul un des deux épipôles entre en jeu et qu'ils ne jouent donc pas un rôle symétrique. Cette équation est donc une forme de factorisation de la matrice fondamentale, mais partielle seulement. Ceci soulève la question suivante : existe-t-il une factorisation de la matrice fondamentale qui ferait explicitement apparaître les deux épipôles et leur ferait jouer un rôle symétrique ? Nous montrons l'existence d'une telle factorisation, qui s'écrit sous la forme $\mathsf{F} \sim [\mathbf{e}']_\wedge \mathsf{G}_{\boldsymbol{\Pi},d} [\mathbf{e}]_\wedge$. Nous montrons que la matrice $\mathsf{G}_{\boldsymbol{\Pi},d}$ représente une extension de la transformation épipolaire au 2D.

Nous distinguons donc deux niveaux de factorisation : la *factorisation partielle* et la *factorisation totale*. La factorisation partielle est basée sur les homographies de plans, tandis que la factorisation totale est basée sur les transformations épipolaires étendues que nous introduisons. Les caractéristiques de ces niveaux de factorisation sont résumées ci-dessous :

niveau de factorisation	entités	transformation impliquée	forme factorisée
factorisation partielle	e', H_{Π}	homographie de plan H_{Π}	$\mathsf{F} \sim [\mathbf{e}']_{\wedge}\mathsf{H}_{\Pi}$
factorisation totale	e, e', $\mathsf{G}_{\Pi,d}$	transformation épipolaire étendue $\mathsf{G}_{\Pi,d}$	$\mathsf{F} \sim [\mathbf{e}']_{\wedge}\mathsf{G}_{\Pi,d}[\mathbf{e}]_{\wedge}$

Ces factorisations permettent une intime compréhension du rôle joué par la matrice fondamentale dans le positionnement relatif de deux caméras non calibrées. La factorisation partielle est à la base de plusieurs réalisations canoniques [90, 130]. La factorisation totale permet de construire plusieurs paramétrisations de la matrice fondamentale [49, 128, 233, 236].

4.2.1 Factorisation partielle

La factorisation partielle de la matrice fondamentale consiste à écrire cette dernière sous la forme suivante :

$$\mathsf{F} \sim [\mathbf{e}']_{\wedge}\mathsf{H}_{\Pi}. \tag{4.1}$$

La transformation H_{Π} est une homographie de plan. Il existe une famille de transformations H_{Π} satisfaisant l'équation (4.1). Cette équation en est une caractérisation indirecte. Notre but est de caractériser directement la famille des homographies de plan par une paramétrisation. Pour ce faire, nous examinons au préalable son rôle dans la factorisation partielle. La figure 4.1 illustre ce rôle ainsi que la paramétrisation.

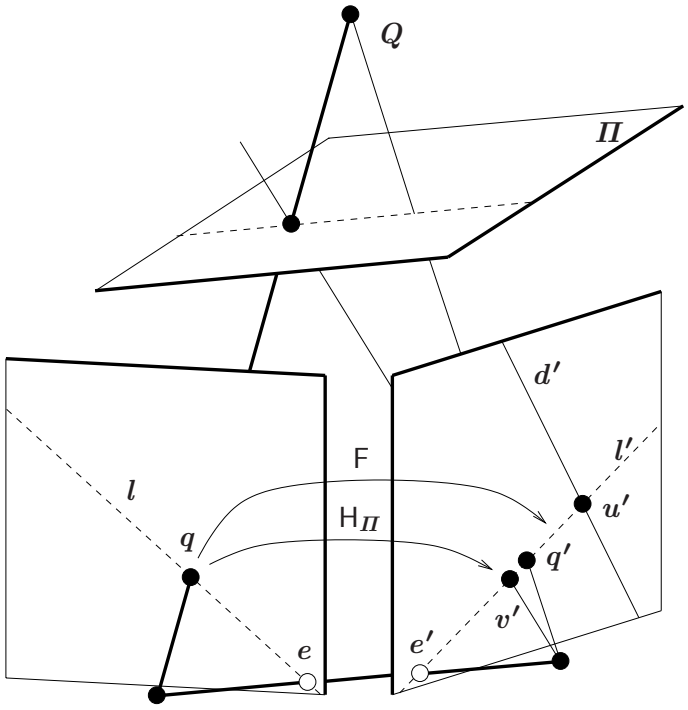


FIG. 4.1 – Illustration du rôle de l'homographie de plan H_{Π} dans la factorisation partielle de la matrice fondamentale $\mathsf{F} \sim [\mathbf{e}']_{\wedge}\mathsf{H}_{\Pi}$ et de la construction d'une paramétrisation de la famille des homographies de plan. Une homographie de plan associe à un point de l'image de gauche un point sur la droite épipolaire correspondante dans l'image de droite. Une sur-paramétrisation en fonction de $\boldsymbol{d}$ et $\bar{\mathbf{\Pi}}$ en est donnée par l'équation $\mathsf{H}_{\Pi} \sim [\mathbf{d}']_{\wedge}\mathsf{F} - \mathbf{e}'\bar{\mathbf{\Pi}}^{\mathsf{T}}$ (voir texte principal). L'homographie de plan H_{Π} est induite par le plan $\boldsymbol{\Pi}$.

Rôle des homographies de plan. *Une homographie de plan H_{Π} associe à un point de l'image de gauche un point sur la droite épipolaire associée dans l'image de droite*. Définissons ce rôle à partir de l'équation (4.1). Pour un point $\boldsymbol{q}$ de l'image de gauche, $\mathsf{F}\mathbf{q} \sim [\mathbf{e}']_{\wedge}\mathsf{H}_{\Pi}\mathbf{q}$ s'interprète comme l'équation de la droite épipolaire $\boldsymbol{l}'$ associée dans l'image de droite. Notons $\mathbf{v}' \sim \mathsf{H}_{\Pi}\mathbf{q}$. Le produit vectoriel $[\mathbf{e}']_{\wedge}\mathbf{v}'$ est l'équation d'une droite

épipolaire car elle contient l'épipôle e'. Cette droite épipolaire est définie par le point v'. Ce dernier doit donc être sur la droite épipolaire l'. Notons que sa position le long de l' n'est pas contrainte. Le rôle d'une homographie de plan est donc d'affecter à un point de l'image de gauche, un point sur la droite épipolaire correspondante dans l'image de droite.

Paramétrisation des homographies de plan. La paramétrisation des homographies de plan est effectuée en construisant un point particulier u' sur l' puis en introduisant une variable α permettant de paramétrer tous les points de l'. Pour ce faire, nous définissons une droite d' quelconque ne contenant pas l'épipôle e' :

expression	interprétation
$\mathbf{q}$	point de l'image de gauche
$\mathbf{l}' \sim \mathsf{F}\mathbf{q}$	droite épipolaire dans l'image de droite
$\mathbf{u}' \sim [\mathbf{d}']_\wedge \mathsf{F}\mathbf{q}$	intersection de la droite épipolaire dans l'image de droite et de la droite d'
$\mathbf{v}' \sim [\mathbf{d}']_\wedge \mathsf{F}\mathbf{q} + \alpha \mathbf{e}'$	point sur la droite épipolaire dans l'image de droite

On observe que cette définition a 3 paramètres libres, l'équation de la droite d' et α. Posons $\alpha = -\bar{\mathbf{\Pi}}^\mathsf{T}\mathbf{q}$ avec $\bar{\mathbf{\Pi}} \in \mathbb{R}^3$. On obtient l'expression suivante des homographies de plan :

$$\mathsf{H}_{\mathit{\Pi},d'} \sim [\mathbf{d}']_\wedge \mathsf{F} - \mathbf{e}'\bar{\mathbf{\Pi}}^\mathsf{T}. \tag{4.2}$$

Dans cette expression, seuls 3 des 5 paramètres $\mathbf{d}'$ et $\bar{\mathbf{\Pi}}$ sont indépendants. Ceci se vérifie facilement en substituant le vecteur $\mathbf{e}'$ par son expression (2.50) dans l'équation (4.2) et en développant. Sans hypothèse sur la position de l'épipôle e', la seule droite d' ne contenant jamais e' est la droite d'équation $\mathbf{d}' \sim \mathbf{e}'$. Ce choix nous donne l'expression de la famille des homographies de plans, équation (2.51), paramétrée par $\bar{\mathbf{\Pi}}$:

$$\mathsf{H}_{\mathit{\Pi}} \sim [\mathbf{e}']_\wedge \mathsf{F} - \mathbf{e}'\bar{\mathbf{\Pi}}^\mathsf{T}. \tag{4.3}$$

L'*homographie de plan canonique* est donnée par $\bar{\mathbf{\Pi}} = \mathbf{0}$:

$$\check{\mathsf{H}} \sim [\mathbf{e}']_\wedge \mathsf{F}. \tag{4.4}$$

Elle est de rang 2 et unique car c'est la seule homographie de plan ayant pour noyaux gauche et droit les épipôles. Cette expression présente une forme de dualité avec la factorisation partielle de la matrice fondamentale donnée par l'équation (4.1).

Luong et Viéville [130] montrent qu'une telle homographie est induite par un plan contenant le deuxième centre de projection et la droite d'équation e'. Ils l'interprètent comme une projection de $\mathbb{P}^2$ vers $\mathbb{P}^1$, car elle affecte à un point de l'image de gauche, un point sur la droite épipolaire correspondante dans l'image de droite, situé sur la droite d'équation e'. Luong et Viéville nomment cette homographie la *projection épipolaire*. On peut généraliser ce concept en partant de l'équation (4.2). Si l'on choisit $\bar{\mathbf{\Pi}} = \mathbf{0}$ dans cette équation, on obtient alors une homographie $[\mathbf{d}']_\wedge \mathsf{F}$. C'est l'homographie d'un plan contenant le deuxième centre de projection et une droite d'équation $\mathbf{d}'$ dans la deuxième image. Par analogie avec le concept de Luong et Viéville, on peut appeler cette homographie la *projection épipolaire généralisée*, dans le sens où elle affecte à tout point de l'image de gauche, un point sur la droite épipolaire correspondante dans l'image de droite et sur la droite d'équation d'.

Propriétés des homographies de plan. Dans le cas général, une homographie de plan met en correspondance les épipôles :

$$\mathbf{e}' \sim \mathsf{H}_{\mathit{\Pi}}\mathbf{e}.$$

Ceci se vérifie facilement en utilisant la paramétrisation (4.2) la plus générale. Elle encapsule la transformation épipolaire, mais seulement la "moitié" des épipôles dans le sens où si un épipôle est donné, $\mathsf{H}_{\mathit{\Pi}}$ permet de retrouver l'autre. La famille des transformations $\mathsf{H}_{\mathit{\Pi}}$ est de dimension 3. D'autres propriétés des homographies de plan sont données en §2.11, chapitre 2.

4.2.2 Factorisation totale

La factorisation totale de la matrice fondamentale consiste à écrire cette dernière sous la forme suivante :

$$\mathsf{F} \sim [\mathbf{e}']_\wedge \mathsf{G}_{\boldsymbol{\Pi},s} [\mathbf{e}]_\wedge. \tag{4.5}$$

La transformation $\mathsf{G}_{\boldsymbol{\Pi},s}$ est ce que nous appelons une *transformation épipolaire étendue*. Notre but est de caractériser la famille des transformations épipolaires étendues $\mathsf{G}_{\boldsymbol{\Pi},s}$ satisfaisant l'équation de factorisation totale (4.5). Pour ce faire, nous examinons au préalable son rôle dans la factorisation totale. La figure 4.2 illustre ce rôle ainsi que la paramétrisation.

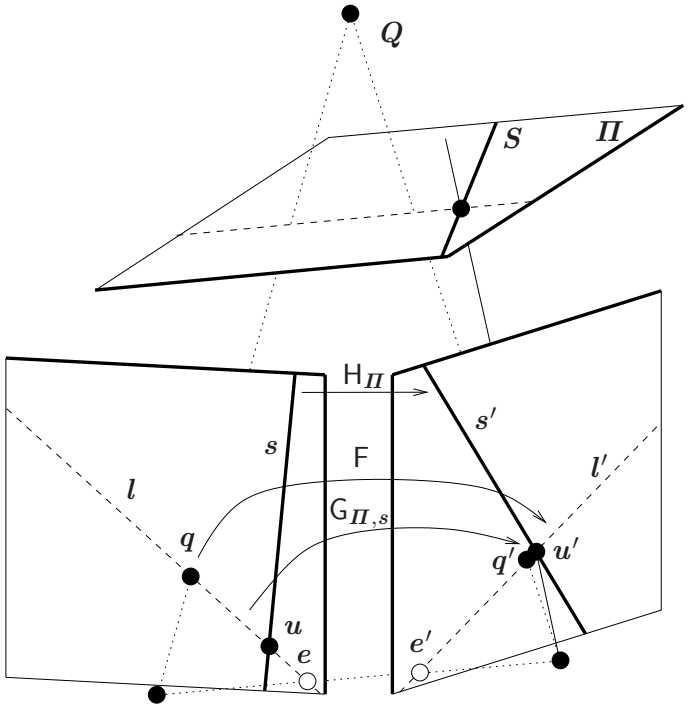


FIG. 4.2 – Illustration du rôle de la transformation épipolaire étendue $\mathsf{G}_{\boldsymbol{\Pi},s}$ dans la factorisation totale de la matrice fondamentale $\mathsf{F} \sim [\mathbf{e}']_\wedge \mathsf{G}_{\boldsymbol{\Pi},s} [\mathbf{e}]_\wedge$ et de la construction de la famille des transformations épipolaires étendues. Une transformation épipolaire étendue associe à une droite épipolaire de l'image de gauche, un point sur la droite épipolaire correspondante dans l'image de droite. Une sur-paramétrisation en fonction de $\boldsymbol{s}$, $\boldsymbol{d}$ et $\bar{\boldsymbol{\Pi}}$ en est donnée par l'équation $\mathsf{G}_{\boldsymbol{\Pi},s,d'} \sim ([\mathbf{d}']_\wedge \mathsf{F} - \mathbf{e}' \bar{\boldsymbol{\Pi}}^\mathsf{T})[\mathbf{s}]_\wedge$ (voir texte principal). La transformation épipolaire étendue $\mathsf{G}_{\boldsymbol{\Pi},s}$ est induite par le plan $\boldsymbol{\Pi}$ et la droite $\boldsymbol{S}$ sur ce plan, dont les projections dans les images sont notées $\boldsymbol{s}$ et $\boldsymbol{s}'$.

Rôle des transformations épipolaires étendues. *Une transformation épipolaire étendue associe à une droite épipolaire de l'image de gauche, un point sur la droite épipolaire correspondante dans l'image de droite*. Elle ne correspond ni à une homographie de plan point-point ni à sa forme duale droite-droite. Examinons l'équation (4.5). Pour un point $\boldsymbol{q}$ de l'image de gauche, $\mathsf{F}\mathbf{q} \sim [\mathbf{e}']_\wedge \mathsf{G}_{\boldsymbol{\Pi},s} [\mathbf{e}]_\wedge \mathbf{q}$ est l'équation de la droite épipolaire $\boldsymbol{l}'$ correspondante dans l'image de droite. D'après le raisonnement de la section précédente, le point $\boldsymbol{u}'$ d'équation $\mathbf{u}' \sim \mathsf{G}_{\boldsymbol{\Pi},s} [\mathbf{e}]_\wedge \mathbf{q}$ doit être un point quelconque sur $\boldsymbol{l}'$. L'équation $[\mathbf{e}]_\wedge \mathbf{q}$ étant celle de la droite épipolaire $\boldsymbol{l}$ associée à $\boldsymbol{q}$ dans l'image de gauche, on en déduit que $\mathsf{G}_{\boldsymbol{\Pi},s}$ associe à une droite épipolaire de l'image de gauche, un point quelconque sur la droite épipolaire correspondante dans l'image de droite. L'application d'une transformation épipolaire étendue à une droite de l'image de gauche qui n'est pas une droite épipolaire n'a pas de sens et n'est pas précisée.

Les transformations épipolaires étendues ont un rôle dual à celui de la matrice fondamentale, dans le sens où la contrainte épipolaire sur les points $\mathbf{q}'^\mathsf{T} \mathsf{F} \mathbf{q} = 0$, se reécrit sur les droites épipolaires via une transformation

épipolaire étendue :

$$\mathbf{l'}^{\mathsf{T}} \mathsf{G}_{\boldsymbol{\Pi},s} \mathbf{l} \;=\; 0.$$

La démonstration de cette égalité est directe. Considérons la contrainte épipolaire et remplaçons la matrice fondamentale par sa factorisation totale (4.5), on obtient :

$$\underbrace{\mathbf{q'}^{\mathsf{T}}[\mathbf{e'}]_\wedge}_{\mathbf{l'}^{\mathsf{T}}} \mathsf{G}_{\boldsymbol{\Pi},s} \underbrace{[\mathbf{e}]_\wedge \mathbf{q}}_{\mathbf{l}} \;=\; 0,$$

où l'on identifie les équations des deux droites épipolaires.

Paramétrisation des transformations épipolaires étendues. Pour construire la paramétrisation recherchée, nous définissons une droite $\boldsymbol{s}$ quelconque ne contenant pas l'épipôle $\boldsymbol{e}$. Soit $\boldsymbol{q}$ un point de l'image de gauche, $\boldsymbol{l'}$ la droite épipolaire correspondante dans l'image de droite et $\boldsymbol{u'}$ un point sur $\boldsymbol{l'}$:

expression	interprétation
$\mathbf{l}$	droite épipolaire dans l'image de gauche
$\mathbf{u} \sim [\mathbf{s}]_\wedge \mathbf{l}$	intersection de la droite épipolaire dans l'image de gauche et de la droite $\boldsymbol{d}$
$\mathbf{u'} \sim \mathsf{H}_{\boldsymbol{\Pi}}[\mathbf{s}]_\wedge \mathbf{l}$	point sur la droite épipolaire dans l'image de droite

On observe que cette définition a 5 paramètres, les 3 de $\mathsf{H}_{\boldsymbol{\Pi}}$ et les 2 de la droite $\boldsymbol{s}$. En remplaçant $\mathsf{H}_{\boldsymbol{\Pi}}$ par son expression (4.2), on obtient l'expression suivante des transformations épipolaires étendues :

$$\mathsf{G}_{\boldsymbol{\Pi},s,d'} \;\sim\; ([\mathbf{d'}]_\wedge \mathsf{F} - \mathbf{e'}\bar{\boldsymbol{\Pi}}^{\mathsf{T}})[\mathbf{s}]_\wedge. \tag{4.6}$$

Cette définition est sur-paramétrée. En choisissant pour $\boldsymbol{d'}$ la seule droite ne contenant jamais $\boldsymbol{e'}$ dont l'équation est $\mathbf{d'} \sim \mathbf{e'}$, on obtient la famille des transformations épipolaires étendues :

$$\mathsf{G}_{\boldsymbol{\Pi},s} \;\sim\; ([\mathbf{e'}]_\wedge \mathsf{F} - \mathbf{e'}\bar{\boldsymbol{\Pi}}^{\mathsf{T}})[\mathbf{s}]_\wedge. \tag{4.7}$$

Sans hypothèse sur la position de l'épipôle $\boldsymbol{e}$ la seule droite $\boldsymbol{s}$ ne contenant jamais $\boldsymbol{e}$ est la droite d'équation $\mathbf{s} \sim \mathbf{e}$. La *transformation épipolaire étendue canonique* est donnée par $\mathbf{s} \sim \mathbf{e}$ et$\bar{\boldsymbol{\Pi}} = \mathbf{0}$:

$$\check{\mathsf{G}} \;\sim\; [\mathbf{e'}]_\wedge \mathsf{F} [\mathbf{e}]_\wedge \;\sim\; \check{\mathsf{H}}[\mathbf{e}]_\wedge. \tag{4.8}$$

Cette expression présente une forme de dualité avec la factorisation totale de la matrice fondamentale donnée par l'équation (4.5). Notons que $\check{\mathsf{G}}$ est de rang 2 et unique car c'est la seule transformation épipolaire étendue ayant pour noyaux gauche et droit les épipôles.

Luong et Viéville [130] introduisent une matrice G définie comme notre transformation épipolaire étendue canonique $\check{\mathsf{G}}$, équation (4.8). Ils montrent la dualité des formules $\mathsf{F} \sim [\mathbf{e'}]_\wedge \mathsf{G} [\mathbf{e}]_\wedge$ et $\mathsf{G} \sim [\mathbf{e'}]_\wedge \mathsf{F} [\mathbf{e}]_\wedge$.

Propriétés des transformations épipolaires étendues. En général, une transformation épipolaire étendue ne contient pas d'information sur les épipôles. Elle n'encapsule donc que la transformation épipolaire, d'où son nom. La famille des transformations épipolaires étendues est de dimension 5. Géométriquement, cela correspond au choix d'un plan et d'une droite sur ce plan. Schmid et Zisserman [173] étudient la famille d'homographies de plan induite par un faisceau de plans autour d'une droite 3D. La transformation épipolaire étendue est de nature différente car elle associe des points alignés au faisceau de droites épipolaires (tout comme une matrice fondamentale associe des droites épipolaires à des points) :

$$\mathsf{G}_{\boldsymbol{\Pi},s} : \mathbb{P}^1 \to \mathbb{P}^1.$$

Les points sont alignés sur la droite $\boldsymbol{s'}$ dont l'équation est donnée par $\mathbf{s'} \sim \mathsf{H}^\star_{\boldsymbol{\Pi}} \mathbf{s}$, où $\mathsf{H}^\star_{\boldsymbol{\Pi}}$ est l'homographie duale (pour les droites) induite par le plan $\boldsymbol{\Pi}$.

La factorisation de Faugeras et Luong. Pour mettre en évidence la transformation épipolaire, Faugeras et Luong [60, §5.2.3] définissent une droite $\boldsymbol{c}$ ne contenant pas l'épipôle $\boldsymbol{e}$ par deux points $\boldsymbol{x}$ et $\boldsymbol{y}$. Ils définissent de même une droite dans la deuxième image. Ils montrent que la matrice fondamentale peut alors s'écrire :

$$\mathsf{F} \sim [\mathbf{e}']_\wedge \overbrace{\begin{pmatrix}\mathbf{x} & \mathbf{y}\end{pmatrix} \underbrace{\begin{pmatrix}a & b\\ c & d\end{pmatrix}}_{\mathsf{g}} \begin{pmatrix}\mathbf{x}'^\mathsf{T}\\ \mathbf{y}'^\mathsf{T}\end{pmatrix}}^{\mathsf{G}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{x}',\boldsymbol{y}'}} [\mathbf{e}]_\wedge. \tag{4.9}$$

Dans cette décomposition, on retrouve la transformation épipolaire g et la transformation épipolaire étendue via la sur-paramétrisation $\mathsf{G}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{x}',\boldsymbol{y}'}$.

4.2.3 Résumé

Nous résumons les deux factorisations de la matrice fondamentale étudiées ci-dessus.

Factorisation partielle : "épipôle + homographie". La factorisation partielle est définie par l'équation $\mathsf{F} \sim [\mathbf{e}']_\wedge \mathsf{H}_{\boldsymbol{\Pi}}$. Le vecteur $\mathbf{e}'$ donne les coordonnées du deuxième épipôle et $\mathsf{H}_{\boldsymbol{\Pi}}$ s'interprète comme une homographie de plan. Sa forme la plus générale est $\mathsf{H}_{\boldsymbol{\Pi},\boldsymbol{d}'} \sim [\mathbf{d}']_\wedge \mathsf{F} - \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T}$. Une paramétrisation minimale est $\mathsf{H}_{\boldsymbol{\Pi}} \sim [\mathbf{e}']_\wedge \mathsf{F} - \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T}$. Le vecteur-3 $\bar{\boldsymbol{\Pi}}$ s'interprète alors comme une équation de plan réduite. L'homographie de plan canonique est $\check{\mathsf{H}} \sim [\mathbf{e}']_\wedge \mathsf{F}$.

Factorisation totale : "épipôle + épipôle + transformation épipolaire étendue". La factorisation totale est définie par l'équation $\mathsf{F} \sim [\mathbf{e}']_\wedge \mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s}} [\mathbf{e}]_\wedge$. Les vecteurs $\mathbf{e}$ et $\mathbf{e}'$ donnent les coordonnées des épipôles et $\mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s}}$ s'interprète géométriquement comme la transformation dégénérée induite par une droite sur un plan et forme la transformation épipolaire étendue. Sa forme la plus générale est $\mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s},\boldsymbol{d}'} \sim ([\mathbf{d}']_\wedge \mathsf{F} - \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T})[\mathbf{s}]_\wedge$. Une paramétrisation minimale est $\mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s}} \sim ([\mathbf{e}']_\wedge \mathsf{F} - \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T})[\mathbf{s}]_\wedge$. Le vecteur-3 $\bar{\boldsymbol{\Pi}}$ s'interprète alors comme une équation de plan réduite et le vecteur-3 $\mathbf{s}$ comme l'équation de la projection dans la première image d'une droite sur ce plan. La transformation épipolaire étendue canonique est $\check{\mathsf{G}} \sim [\mathbf{e}']_\wedge \mathsf{F} [\mathbf{e}]_\wedge$.

4.3 Paramétrisation de la matrice fondamentale

Nous étudions l'état de l'art de la paramétrisation de la matrice fondamentale. Les méthodes étudiées et leurs caractéristiques sont résumées dans le tableau ci-dessous.

méthode	# paramètres	# cartes	caractéristiques	références	décrit en
Luong et Faugeras	8	1	épipôles finis	[128]	§4.3.1
Luong et Faugeras	7	4	épipôles finis	[128]	§4.3.1
Zhang	8	9	aucune	[233]	§4.3.2
Csurka et al.	7	36	aucune	[49]	§4.3.2
Zhang et Loop	7	1	fonction de coût modifiée	[236]	§4.3.3
Torr et Zisserman	28	-	mise à jour minimale	[206]	§4.3.4
Hartley	12	1	aucune	[90]	§4.3.5

Le formalisme de factorisation totale de la matrice fondamentale que nous avons introduit permet de construire simplement les 4 premières paramétrisations de ce tableau, car elles sont basées sur une représentation de la géométrie épipolaire en épipôles et transformation épipolaire. Nous proposons deux contructions pour ces paramétrisations, une construction géométrique basée sur la transformation épipolaire étendue, et une construction algébrique. Dans le cas de la paramétrisation de Luong et Faugeras, nous proposons une réalisation particulière de la matrice fondamentale exploitant l'hypothèse d'épipôles finis. La méthode de Zhang et Loop est basée sur la paramétrisation de Luong et Faugeras. La dernière paramétrisation, celle de Hartley, est basée sur la

factorisation partielle, c'est à dire la représentation de la matrice fondamentale par une homographie 2D et un épipôle. Des expérimentations concernant ces méthodes sont décrites en §4.5. La méthode proposée par Torr et Zisserman [206, 211] est basée sur une représentation de la matrice fondamentale par 7 correspondances de points. Ce sont les coordonnées de ces points qui sont mises à jour lors de l'optimisation non-linéaire. Elle est particulière dans le sens où la mise en correspondance entre les paramètres et la matrice fondamentale est très indirecte (elle nécessite la résolution d'un système linéaire et d'une équation cubique).

4.3.1 La paramétrisation de Luong et Faugeras

Luong et Faugeras [128] se concentrent sur le cas d'épipôles n'étant pas à l'infini, c'est à dire pas "trop loin" de l'image, ce qui correspond à des caméras dont les plans image ne sont pas parallèles à la droite reliant les centres de projection. Les épipôles peuvent alors s'écrire en coordonnées affines : $\mathbf{e}^{\mathsf{T}} \sim (e_1 \;\; e_2 \;\; 1)$ et $\mathbf{e}'^{\mathsf{T}} \sim (e'_1 \;\; e'_2 \;\; 1)$. Ils sont naturellement paramétrés de manière minimale par les coefficients e_1, e_2, e'_1 et e'_2. Pour compléter cette paramétrisation, il reste à traiter la transformation épipolaire. La difficulté est de faire apparaître 4 coefficients représentant cette transformation, voir ci-dessous.

Cette paramétrisation a 8 paramètres et ne nécessite qu'une seule carte. Nous en donnons une construction géométrique et algébrique. Les 4 coefficients de la transformation épipolaire sont définis à un facteur multiplicatif près et fixent l'échelle de la matrice fondamentale. La paramétrisation n'est donc minimale que sous réserve de la gestion de l'échelle de ces 4 coefficients. Luong et Faugeras proposent de normaliser par le plus important. Cela revient à utiliser 4 cartes. La meilleure carte se choisit en sélectionnant le coefficient dont la valeur absolue est la plus élevée.

La construction géométrique est basée sur la factorisation totale de la matrice fondamentale. La construction algébrique repose sur une élimination analytique de coefficients de la matrice fondamentale, basé sur les équations $\mathsf{F}\mathbf{e} = \mathsf{F}^{\mathsf{T}}\mathbf{e}' = \mathbf{0}$.

4.3.1.1 Construction géométrique

Pour faire apparaître la transformation épipolaire, considérons l'expression générale (4.6) des transformations épipolaires étendues :

$$\mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s},\boldsymbol{d}'} \;\sim\; ([\mathbf{d}']_\wedge \mathsf{F} - \mathbf{e}'\bar{\mathbf{\Pi}}^{\mathsf{T}})[\mathbf{s}]_\wedge.$$

Rappelons que $\boldsymbol{s}$ et $\boldsymbol{d}'$ s'interprètent comme des droites de $\mathbb{P}^2$, telles que $\boldsymbol{e} \notin \boldsymbol{s}$ et $\boldsymbol{e}' \notin \boldsymbol{d}'$. Les épipôles étant supposés ne pas être à l'infini, Luong et Faugeras fixent $\mathbf{s} \sim \mathbf{d}' \sim \mathbf{l}_\infty \sim (0 \;\; 0 \;\; 1)^{\mathsf{T}}$. L'expression suivante est alors obtenue pour $\bar{\mathbf{\Pi}} = \mathbf{0}$:

$$\begin{aligned} \mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s},\boldsymbol{d}'} &\sim [\mathbf{l}_\infty]_\wedge \mathsf{F} [\mathbf{l}_\infty]_\wedge \\ &\sim \begin{pmatrix} -f_{22} & f_{21} & 0 \\ f_{12} & -f_{11} & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{aligned} \tag{4.10}$$

On voit apparaître dans cette expression les 4 coefficients de la transformation épipolaire : f_{11}, f_{12}, f_{21} et f_{22}. En substituant l'expression ci-dessus dans la factorisation totale de la matrice fondamentale (4.5), nous obtenons :

$$\begin{aligned} \mathsf{F} &\sim [\mathbf{e}']_\wedge \mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s},\boldsymbol{d}'} [\mathbf{e}]_\wedge \\ &\sim \begin{pmatrix} f_{11} & f_{12} & -e_1 f_{11} - e_2 f_{12} \\ f_{21} & f_{22} & -e_1 f_{21} - e_2 f_{22} \\ -e'_1 f_{11} - e'_2 f_{21} & -e'_1 f_{12} - e'_2 f_{22} & e_1 e'_1 f_{11} + e_1 e'_2 f_{21} + e_2 e'_1 f_{12} + e_2 e'_2 f_{22} \end{pmatrix}. \end{aligned} \tag{4.11}$$

La carte ainsi construite associe les 8 paramètres $(e_1, e_2, e'_1, e'_2, f_{11}, f_{12}, f_{21}, f_{22})$ aux coefficients de la matrice fondamentale. Cette carte garantit que la contrainte de rang 2 est satisfaite.

Notons que cette paramétrisation se construit de même à partir de la sur-paramétrisation de la transformation épipolaire étendue de Faugeras et Luong donnée par l'équation (4.9), en choisissant des coordonnées canoniques pour les points $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{x}'$ et $\boldsymbol{y}'$. Rappelons que les points $\boldsymbol{x}$ et $\boldsymbol{y}$ doivent définir une droite ne contenant pas l'épipôle $\boldsymbol{e}$, de même pour les points $\boldsymbol{x}'$, $\boldsymbol{y}'$ et l'épipôle $\boldsymbol{e}'$. Faugeras et Luong [60, §5.2.3] choisissent $\mathbf{x}^\mathsf{T} \sim \mathbf{x}'^\mathsf{T} \sim (1\ 0\ 0)$ et $\mathbf{y}^\mathsf{T} \sim \mathbf{y}'^\mathsf{T} \sim (0\ 1\ 0)$. Ils obtiennent alors une expression de la transformation épipolaire étendue identique à celle obtenue ci-dessus, équation (4.10), à partir de notre paramétrisation.

4.3.1.2 Construction algébrique

La paramétrisation de Luong et Faugeras se construit de manière algébrique comme suit. L'idée est d'utiliser le fait que $\mathbf{e}$ et $\mathbf{e}'$ sont les noyaux droit et gauche respectivement de la matrice fondamentale. Soient $\mathbf{c}_i$, $i \in \{1 \dots 3\}$ les colonnes de F. L'équation suivante est vérifiée :

$$\mathsf{F}\mathbf{e} \;=\; e_1\mathbf{c}_1 + e_2\mathbf{c}_2 + \mathbf{c}_3 \;=\; \mathbf{0}.$$

On obtient donc :

$$\mathbf{c}_3 \;=\; -e_1\mathbf{c}_1 - e_2\mathbf{c}_2,$$

et l'écriture suivante de la matrice fondamentale :

$$\mathsf{F} \;\sim\; \begin{pmatrix}\mathbf{c}_1 & \mathbf{c}_2 & -e_1\mathbf{c}_1 - e_2\mathbf{c}_2\end{pmatrix}.$$

Si l'on répète une opération similaire sur les lignes de F, en utilisant l'équation $\mathsf{F}^\mathsf{T}\mathbf{e}' = \mathbf{0}$, on obtient :

$$\mathsf{F} \;\sim\; \begin{pmatrix} f_{11} & f_{12} & -e_1 f_{11} - e_2 f_{12} \\ f_{21} & f_{22} & -e_1 f_{21} - e_2 f_{22} \\ -e'_1 f_{11} - e'_2 f_{21} & -e'_1 f_{12} - e'_2 f_{22} & e_1 e'_1 f_{11} + e_1 e'_2 f_{21} + e_2 e'_1 f_{12} + e_2 e'_2 f_{22} \end{pmatrix}.$$

Cette carte est la même que celle obtenue précédemment, équation (4.11).

4.3.1.3 Paramétrisation minimale

La carte (4.11) décrit une association entre 8 paramètres et la matrice fondamentale. La paramétrisation n'est donc pas minimale. En effet, l'échelle de la transformation épipolaire et donc celle de la matrice fondamentale correspondante donnée par la carte (4.11) n'est pas fixée. Luong et Faugeras proposent de normaliser par le plus important des 4 coefficients de la transformation épipolaire. Il y a 4 possibilités, d'où la nécessité de 4 cartes. La meilleure carte se choisit en sélectionnant le coefficient dont la valeur absolue est la plus élevée. Par exemple, lorsque la valeur absolue de f_{11} est supérieure à celles de f_{12}, f_{21} et f_{22}, les paramètres sont $(e_1, e_2, e'_1, e'_2, f_{12}, f_{21}, f_{22})$ et la carte associée est :

$$\mathsf{F} \;\sim\; \begin{pmatrix} 1 & f_{12} & -e_1 - e_2 f_{12} \\ f_{21} & f_{22} & -e_1 f_{21} - e_2 f_{22} \\ -e'_1 - e'_2 f_{21} & -e'_1 f_{12} - e'_2 f_{22} & e_1 e'_1 + e_1 e'_2 f_{21} + e_2 e'_1 f_{12} + e_2 e'_2 f_{22} \end{pmatrix}. \tag{4.12}$$

4.3.1.4 Réalisation

Nous proposons de réaliser la matrice fondamentale sous l'hypothèse d'épipôles finis de Luong et Faugeras. Il est bien sûr possible d'utiliser une des bases canoniques précédemment évoquées, mais l'hypothèse d'épipôles finis ne serait alors pas exploitée. Nous cherchons une réalisation de la forme donnée par les équations (2.43) et (2.44), ce qui implique le choix d'une homographie de référence :

$$\mathsf{P} \;\sim\; (\ \mathrm{I}\quad \mathbf{0}) \quad \text{et} \quad \mathsf{P}' \;\sim\; (\ \mathsf{H}_r \quad \mathbf{e}').$$

Pour choisir cette homographie en exploitant pleinement l'hypothèse d'épipôles finis, nous utilisons le formalisme de la factorisation partielle de la matrice fondamentale. L'équation (4.2) paramétrise la famille des homographies de plan :

$$\mathsf{H}_{\boldsymbol{\Pi},\mathbf{d}'} \sim [\mathbf{d}']_\wedge \mathsf{F} + \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T}.$$

Dans cette expression, $\mathbf{d}'$ est l'équation d'une droite ne contenant pas l'épipôle $\boldsymbol{e}'$. Comme précédemment, l'hypothèse d'épipôles finis permet de choisir $\mathbf{d}' \sim \mathbf{l}_\infty \sim (0\ 0\ 1)^\mathsf{T}$. Nous obtenons alors, pour $\bar{\boldsymbol{\Pi}} = \mathbf{0}$:

$$\begin{aligned} \mathsf{P}' &\sim \begin{pmatrix} [\mathbf{l}_\infty]_\wedge \mathsf{F} & \mathbf{e}' \end{pmatrix} \\ &\sim \begin{pmatrix} -f_{21} & -f_{22} & -f_{23} & e'_1 \\ f_{11} & f_{12} & f_{13} & e'_2 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \end{aligned} \tag{4.13}$$

Notons que cette caméra est de forme affine.

4.3.2 Les paramétrisations de Zhang et de Csurka et al.

Les paramétrisations de Csurka et al. [49] et Zhang [233] généralisent celle de Luong et Faugeras dans le sens où l'hypothèse d'épipôles finis est levée. L'utilisation de plusieurs cartes est nécessaire. De manière similaire à la paramétrisation de Luong et Faugeras, nous en donnons une construction géométrique et algébrique. La paramétrisation de Zhang est basée sur 9 cartes et 8 paramètres. Elle ne fixe pas l'échelle de la matrice fondamentale. La paramétrisation de Csurka et al. fixe cette échelle et n'utilise que 7 paramètres, mais un total de 36 cartes. Nous examinons la paramétrisation des épipôles puis les constructions algébriques et géométriques des 9 cartes associées, ce qui donne la paramétrisation de Zhang. Enfin, similairement au cas de la paramétrisation de Luong et Faugeras, la normalisation de la matrice fondamentale réduit le nombre de paramètres à 7 et augmente le nombre de cartes par un facteur 4. Nous obtenons ainsi la paramétrisation de Csurka et al. Finalement, nous examinons un critère de choix pour la meilleure carte.

4.3.2.1 Paramétrisation des épipôles

Contrairement au cas de Luong et Faugeras où il est naturel de paramétrer les épipôles en termes de leurs coordonnées affines, il faut dans le cas général utiliser des cartes. Considérons l'épipôle $\boldsymbol{e}$. Le vecteur-3 $\mathbf{e}$ qui le représente est défini à un facteur près. Il faut donc normaliser par un de ses coefficients, et l'éliminer de la paramétrisation. Ce coefficient ne doit pas être nul ou trop "petit". On choisit donc le coefficient ayant la valeur absolue la plus élevée. Soit m l'indice de ce coefficient. Il existe 3 possibilités et 3 cartes sont donc mises en jeu pour la paramétrisation de $\mathbf{e}$. Par exemple, si $m = 2$, le jeu de paramètres est (e_1, e_3) et la carte correspondante est $\mathbf{e}^\mathsf{T} \sim (e_1\ 1\ e_3)$. Un raisonnement similaire pour l'épipôle $\boldsymbol{e}'$ permet de conclure qu'un indice m' est nécessaire et que $3 \times 3 = 9$ cartes sont nécessaires pour la paramétrisation des épipôles. Les 9 cartes de la matrice fondamentale sont construites ci-dessous.

4.3.2.2 Construction géométrique

Comme dans le cas de Luong et Faugeras, cette construction est basée sur la factorisation totale de la matrice fondamentale, équation (4.5). Considérons l'expression (4.6) des transformations épipolaires étendues :

$$\mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s},\mathbf{d}'} \sim ([\mathbf{d}']_\wedge \mathsf{F} + \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T})[\mathbf{s}]_\wedge.$$

Le choix inconditionnel $\mathbf{s} \sim \mathbf{d}' \sim \mathbf{l}_\infty$ n'est plus possible car on n'a pas de connaissance a priori sur la position des épipôles. Il est nécessaire de se servir des indices m et m'. Ces indices garantissent $e_m \neq 0$ et $e'_{m'} \neq 0$. Par exemple, dans le cas $m = m' = 3$, la carte de la matrice fondamentale est celle de Luong et Faugeras, équation (4.11).

Examinons le choix de $\boldsymbol{s}$ pour $m \neq 3$. Lorsque $m = 1$, $e_1 \neq 0$ et par suite, l'épipôle $\boldsymbol{e}$ ne se trouve pas sur l'axe des y dans l'image. Cet axe, d'équation $\mathbf{l}_y^\mathsf{T} \sim (1\ 0\ 0)$, peut être choisi comme droite $\boldsymbol{s}$. Lorsque $m = 2$, on montre de manière similaire que $\mathbf{l}_x^\mathsf{T} \sim (0\ 1\ 0)$ est un choix possible pour la droite $\boldsymbol{s}$.

Construisons une carte de la matrice fondamentale, par exemple pour $m = 2$ et $m' = 1$, c'est à dire $\mathbf{s} \sim \mathbf{l}_x$ et $\mathbf{d}' \sim \mathbf{l}_y$. La transformation épipolaire étendue s'écrit, avec $\bar{\boldsymbol{\Pi}} = \mathbf{0}$:

$$\begin{aligned} \mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s},\boldsymbol{d}'} &\sim [\mathbf{l}_y]_\wedge \mathsf{F} [\mathbf{l}_x]_\wedge \\ &\sim \begin{pmatrix} 0 & 0 & 0 \\ f_{32} & 0 & -f_{31} \\ -f_{22} & 0 & f_{21} \end{pmatrix}, \end{aligned}$$

où l'on voit apparaître 4 coefficients pour la transformation épipolaire, f_{21}, f_{22}, f_{31} et f_{32}. La carte de la matrice fondamentale est obtenue en substituant l'expression ci-dessus dans l'équation (4.5), avec $\mathbf{e}^\mathsf{T} \sim (e_1\ 1\ e_3)$ et $\mathbf{e}'^\mathsf{T} \sim (1\ e'_2\ e'_3)$:

$$\begin{aligned} \mathsf{F} &\sim [\mathbf{e}']_\wedge \mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s},\boldsymbol{d}'} [\mathbf{e}]_\wedge \\ &\sim \begin{pmatrix} -e'_3 f_{31} - e'_2 f_{21} & e_3 e'_3 f_{32} + e_3 e'_2 f_{22} + e_1 e'_3 f_{31} + e_1 e'_2 f_{21} & -e'_3 f_{32} - e'_2 f_{22} \\ f_{21} & -e_3 f_{22} - e_1 f_{21} & f_{22} \\ f_{31} & -e_3 f_{32} - e_1 f_{31} & f_{32} \end{pmatrix}. \end{aligned}$$

La carte ainsi construite associe les 8 paramètres $(e_1, e_3, e'_2, e'_3, f_{21}, f_{22}, f_{31}, f_{32})$ à la matrice fondamentale. Cette carte garantit que la contrainte de rang 2 est satisfaite.

Romano [168] montre que ces cartes peuvent être construites à partir de la sur-paramétrisation de Faugeras et Luong de la transformation épipolaire étendue $\mathsf{G}_{\boldsymbol{x},\boldsymbol{y},\boldsymbol{x}',\boldsymbol{y}'}$ donnée par l'équation (4.10). Ceci est possible si des coordonnées canoniques, c'est à dire parmi les 3 vecteurs $(1\ 0\ 0)^\mathsf{T}$, $(0\ 1\ 0)^\mathsf{T}$ et $(0\ 0\ 1)^\mathsf{T}$, sont affectées aux points $\boldsymbol{x}$, $\boldsymbol{y}$, $\boldsymbol{x}'$ et $\boldsymbol{y}'$. Elle montre de même que certaines cartes peuvent être rendues numériquement plus stables en choisissant d'autres types de coordonnées pour ces points.

4.3.2.3 Construction algébrique

Soient $\mathbf{c}_i$, $i \in \{1 \dots 3\}$ les colonnes de F. Considérons l'équation $\mathsf{F}\mathbf{e} = \mathbf{0}$, qui s'écrit aussi :

$$e_1 \mathbf{c}_1 + e_2 \mathbf{c}_2 + e_3 \mathbf{c}_3 = \mathbf{0}.$$

L'indice m défini dans la paramétrisation des épipôles indique un coefficient de $\mathbf{e}$ non nul. On peut donc exprimer la colonne $\mathbf{c}_m$ comme :

$$\mathbf{c}_m = -\frac{\sum_{i=1, i\neq m}^{3} e_i \mathbf{c}_i}{e_m}.$$

De même, lors de l'utilisation de la contrainte $\mathsf{F}^\mathsf{T}\mathbf{e}' = \mathbf{0}$ sur les lignes de F, nous rencontrons trois possibilités de choix pour m', l'indice d'un coefficient non nul de $\mathbf{e}'$. Si l'on ne fixe pas l'échelle de la matrice fondamentale, on arrive à un total de $3 \times 3 = 9$ cartes différentes correspondant aux choix de m et m', avec un jeu de 8 paramètres. Les cartes ainsi formées sont les mêmes que celles obtenues dans la construction géométrique.

4.3.2.4 Paramétrisation minimale

La paramétrisation précédente est basée sur 9 cartes et 8 paramètres. Elle ne fixe pas l'échelle de la transformation épipolaire et donc celle de la matrice fondamentale. Les 4 coefficients de la transformation épipolaire sont mis en évidence par les cartes utilisées. Pour fixer l'échelle de cette transformation, on peut, comme Luong et Faugeras, normaliser par le coefficient de plus grande valeur absolue, ce qui donne 4 possibilités. Les coefficients de la transformation épipolaire étant différents pour chacune des 9 cartes précédemment construites, on arrive finalement à un total de $3 \times 3 \times 4 = 36$ cartes différentes et 7 paramètres. C'est la méthode proposée par Csurka et al. [49].

4.3.2.5 Choix de la meilleure carte

Il y a plusieurs stratégies pour choisir la carte la plus adaptée à une matrice fondamentale donnée. Zhang, ainsi que Csurka et al. proposent de maximiser le rang de la matrice Jacobienne associant la matrice fondamentale à ses paramètres, qui est équivalent à la norme d'un vecteur contenant les déterminants des sous-matrices. Ces normes ont des expressions simples en fonction des paramètres.

Un autre manière de sélectionner la meilleure carte est de procéder entité par entité : les coefficients ayant la valeur absolue maximale des coordonnées des épipôles, puis de la transformation épipolaire forment 36 combinaisons.

4.3.3 La solution de Zhang et Loop par transformation projective des images

Zhang et Loop [236] proposent de transformer les points image. Une transformation projective est calculée pour chaque image. Ces transformations garantissent que dans les images transformées, les épipôles ne sont pas à l'infini, et que le coefficient de plus grande valeur absolue de la matrice fondamentale est f_{11}. La carte de Luong et Faugeras donnée par l'équation (4.12), avec le jeu de paramètres $(e_1, e_2, e'_1, e'_2, f_{12}, f_{21}, f_{22})$, peut alors être utilisée. L'avantage de cette approche est qu'une seule carte est nécessaire. Par contre, les transformations projectives utilisées ne préservent pas le modèle de bruit des points image originaux. Or les fonctions de coût utilisées, en particulier l'erreur de reprojection, n'ont un sens statistique que si le modèle de bruit original est préservé. Zhang et Loop considèrent le critère du gradient pondéré. Ils proposent une pondération des équations permettant de préserver le modèle de bruit original au premier ordre pour ce critère.

4.3.4 La solution de Torr et Zisserman par correspondances de points

La méthode proposée par Torr et Zisserman [206, 211] est basée sur une représentation de la matrice fondamentale par 7 correspondances de points. Notons qu'ils appliquent de même cette idée à la paramétrisation de différentes relations géométriques, par exemple les homographies de plan. Ce sont les coordonnées des points qui sont mises à jour lors de l'optimisation non-linéaire. La fonction de coût est estimée en calculant la matrice fondamentale correspondante. Il n'y a pas de solution analytique simple pour ce calcul[2]. Cette paramétrisation n'est pas minimale. Torr et Zisserman donnent une solution pour effectuer l'estimation de manière minimale, consistant à ne bouger qu'une seule coordonnée d'un point de chaque correspondance. On arrive alors à 7 paramètres. Cette solution présente des singularités, par exemple changer la coordonnées x d'un point si la droite épipolaire correspondante est horizontale ne produit pas d'effet sur la géométrie épipolaire représentée. Pour palier à ce défaut, Torr et Zisserman considèrent la variété correspondant à l'estimation courante de la matrice fondamentale dans leur espace de paramètres et proposent de perturber leur représentation orthogonalement à cette variété.

4.3.5 La paramétrisation "épipôle + homographie"

Hartley [90] propose d'utiliser la factorisation partielle (4.1) et utilise comme paramètres les coefficients d'une homographie 2D $\mathsf{H}_{\boldsymbol{\Pi}}$ et du deuxième épipôle $\mathbf{e}'$. La matrice fondamentale est donnée par $\mathsf{F} \sim [\mathbf{e}']_\wedge \mathsf{H}_{\boldsymbol{\Pi}}$. Cette paramétrisation a 12 paramètres. Elle se situe à la frontière entre les paramétrisations basées sur la matrice fondamentale et celles basées sur les matrices de projection car la deuxième matrice de projection de la réalisation canonique est directement donnée par $\mathsf{P}' \sim (\ \mathsf{H}_{\boldsymbol{\Pi}} \quad \mathbf{e}')$.

4.4 La représentation orthonormale de la matrice fondamentale

Cette section traite de la *représentation orthonormale* de la matrice fondamentale que nous proposons et de son utilisation pour l'estimation non-linéaire. Les paramétrisations existantes sont globales, et nous avons

[2] L'estimation de la matrice fondamentale à partir de 7 correspondances de points est basée sur la résolution d'une système linéaire puis d'une équation cubique. Cette dernière peut avoir 1 ou 3 solutions réelles, Torr et Zisserman retiennent celle qui minimise la fonction de coût.

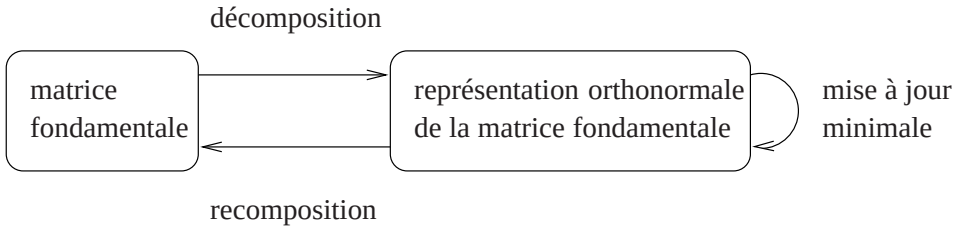


FIG. 4.3 – Principe d'utilisation de la représentation orthonormale de la matrice fondamentale pour l'optimisation non-linéaire. La représentation orthonormale est une sur-paramétrisation globale pouvant être mise à jour localement de manière minimale.

vu que cela entraînait soit une sur-paramétrisation, soit l'utilisation de plusieurs cartes. Notre idée est d'utiliser une sur-paramétrisation globale, la représentation orthonormale, associée à une paramétrisation minimale locale, utilisée pour mettre à jour la sur-paramétrisation globale à chaque étape de l'estimation non-linéaire. Le principe d'optimisation est illustré sur la figure 4.3. Le vocabulaire suivant est employé. La *décomposition* désigne le passage de la matrice fondamentale à sa représentation orthonormale et la *recomposition* la réciproque. Nous examinons les opérations de décomposition et de recomposition et la consistance et complétude de la représentation orthonormale. Nous étudions un schéma de mise à jour minimale de la représentation orthonormale, ainsi que ses propriétés de consistance et de complétude. Finalement, nous étudions l'utilisation de la représentation orthonormale pour l'estimation non-linéaire de la matrice fondamentale, par minimisation de la somme des carrés des distances entre les points et les droites épipolaires (critère de l'approche réduite), puis par minimisation de l'erreur de reprojection (critère de l'approche directe). Ces résultats sont résumés dans le tableau 4.1 sur la page 69.

4.4.1 Décomposition

Soit F une matrice fondamentale donnée, estimée avec par exemple l'algorithme des huit points. Considérons sa décomposition en valeurs singulières :

$$\mathsf{F} \quad \sim \quad \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}. \tag{4.14}$$

Dans cette équation, U et V sont des matrices orthonormales de taille (3×3) et Σ est une matrice diagonale contenant les valeurs singulières de F. Les épipôles gauche et droit sont donnés respectivement par la dernière colonne de V et U : $\mathbf{e} \sim \mathbf{v}_3$ et $\mathbf{e}' \sim \mathbf{u}_3$. Comme F est une matrice de rang 2, sa dernière valeur singulière est nulle. Nous pouvons donc écrire $\Sigma \sim \mathrm{diag}(\sigma_1, \sigma_2, 0)$, où $\sigma_1 \geq \sigma_2 > 0$. De même, la première valeur singulière est forcement non nulle, ce qui permet d'écrire $\sigma = \sigma_2/\sigma_1$ et :

$$\Sigma \quad \sim \quad \mathrm{diag}(1, \sigma, 0).$$

Cette factorisation montre que toute matrice fondamentale peut être représentée par un triplet $(\mathsf{U}, \mathsf{V}, \sigma)$, c'est à dire deux matrices orthonormales de $O(3)$ et un scalaire de $\mathbb{R}^*$. Nous appelons ce triplet la *représentation orthonormale* de la matrice fondamentale. Dans le cas général, cette représentation a des ambiguïtés discrètes, par exemple, $\mathsf{F} \sim \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}} \sim (-\mathsf{U})\Sigma\mathsf{V}^{\mathsf{T}}$. Notons que dans le cas d'une matrice essentielle, $\sigma = 1$ et la représentation orthonormale a une ambiguïté de dimension 2. Ce cas est examiné en détails en section 4.4.8.

4.4.2 Recomposition

Etant donné un triplet $(\mathsf{U}, \mathsf{V}, \sigma)$, la matrice fondamentale correspondante est obtenue en recomposant la décomposition en valeurs singulières, équation (4.14), ce qui donne l'équation de *recomposition* :

$$\mathsf{F} \quad \sim \quad \mathbf{u}_1\mathbf{v}_1^{\mathsf{T}} + \sigma\mathbf{u}_2\mathbf{v}_2^{\mathsf{T}}, \tag{4.15}$$

où les $\mathbf{u}_i$ et les $\mathbf{v}_i$ sont les colonnes des matrices U et V respectivement.

4.4.3 Consistance et complétude

4.4.3.1 Consistance

Nous examinons les contraintes de consistance devant être satisfaites par un triplet $(\mathsf{U}, \mathsf{V}, \sigma)$. Si ces contraintes ne sont pas respectées, la loi de recomposition donnée par l'équation (4.15) garantit tout de même que la matrice F obtenue est une matrice fondamentale. L'intérêt du respect de ces contraintes est de garantir globalement l'unicité de la représentation orthonormale. Par construction, U et V doivent être des matrices orthonormales. Les équations suivantes doivent donc être vérifiées :

$$\mathsf{U}^\mathsf{T}\mathsf{U} \;=\; \mathsf{V}^\mathsf{T}\mathsf{V} \;=\; \mathrm{I}_{(3\times 3)}. \tag{4.16}$$

Ces contraintes de consistance ne laissent que 7 degrés de liberté au triplet $(\mathsf{U}, \mathsf{V}, \sigma)$. Le scalaire σ est obtenu comme $\sigma = \sigma_2/\sigma_1$ avec $\sigma_1 \geq \sigma_2 > 0$, d'où les bornes suivantes :

$$1 \;\geq\; \sigma \;>\; 0. \tag{4.17}$$

4.4.3.2 Complétude

La représentation orthonormale est complète dans le sens où toute matrice fondamentale, c'est à dire toute matrice de rang 2 définie à un facteur prés, peut être décomposée comme proposée ci-dessus.

4.4.4 Réalisation

Un avantage de la représentation orthonormale est que la réalisation canonique de la matrice fondamentale représentée s'écrit de manière directe et simple. Considérons la *réalisation canonique singulière* correspondant aux matrices de projection données par les équations (2.43) et (2.45) :

$$\begin{aligned} \mathsf{P} &\sim (\ \mathrm{I} \quad \mathbf{0}) \\ \check{\mathsf{P}}' &\sim (\ [\mathbf{e}']_\wedge \mathsf{F} \quad \mathbf{e}'). \end{aligned}$$

Les coordonnées de l'épipôle $\boldsymbol{e}'$ sont données par la $3^{\text{ème}}$ colonne de U et la matrice fondamentale par l'équation de recomposition (4.15) :

$$\check{\mathsf{P}}' \;\sim\; (\ [\mathbf{u}_3]_\wedge(\mathbf{u}_1\mathbf{v}_1^\mathsf{T} + \sigma\mathbf{u}_2\mathbf{v}_2^\mathsf{T}) \quad \mathbf{u}_3).$$

Comme U est une matrice orthonormale, $[\mathbf{u}_3]_\wedge\mathbf{u}_1 = \pm\mathbf{u}_2$ et $[\mathbf{u}_3]_\wedge\mathbf{u}_2 = \mp\mathbf{u}_1$, ce qui donne :

$$\check{\mathsf{P}}' \;\sim\; (\ \mathbf{u}_2\mathbf{v}_1^\mathsf{T} - \sigma\mathbf{u}_1\mathbf{v}_2^\mathsf{T} \quad \mathbf{u}_3). \tag{4.18}$$

De manière similaire, on peut écrire la *réalisation canonique non singulière*, dont la deuxième matrice de projection est donnée par l'équation (2.47) :

$$\mathcal{P}' \;\sim\; (\ \mathbf{u}_2\mathbf{v}_1^\mathsf{T} - \sigma\mathbf{u}_1\mathbf{v}_2^\mathsf{T} + \mathbf{u}_3\mathbf{v}_3^\mathsf{T} \quad \mathbf{u}_3).$$

4.4.5 Optimisation minimale

Nous avons montré que toute matrice fondamentale pouvait être représentée par un triplet $(\mathsf{U}, \mathsf{V}, \sigma)$, nommé représentation orthonormale. Nous cherchons un moyen de parcourir la variété des matrices fondamentales via la représentation orthonormale. Ceci consiste à trouver une loi ou schéma de mise à jour de la représentation orthonormale, permettant de se déplacer sur la variété des matrice fondamentales en respectant les contraintes de consistance (4.16) et (4.17). Nous voulons de plus que cette mise à jour soit minimale, c'est à dire qu'elle ne dépende que de 7 paramètres. Nous proposons la solution suivante, inspirée de la mise à jour des matrices de rotation 3D de la section 2.7.3, chapitre 2, pour U et V et par mise à jour additive standard de σ. Soit F_k

l'estimation de la matrice fondamentale à l'itération k et $(\mathsf{U}_k, \mathsf{V}_k, \sigma_k)$ sa représentation orthonormale. Avec des notations similaires pour F_{k+1} le schéma de mise à jour s'écrit :

$$\left\{ \begin{array}{rcl} \mathsf{U}_{k+1} & = & \mathsf{U}_k \mathsf{R}_{3D}(\mathbf{u}_k) \\ \mathsf{V}_{k+1} & = & \mathsf{V}_k \mathsf{R}_{3D}(\mathbf{v}_k) \\ \sigma_{k+1} & = & \sigma_k + \delta_k, \end{array} \right. \tag{4.19}$$

où $\mathbf{u}_k$ et $\mathbf{v}_k$ sont deux vecteurs de taille (3×1). L'optimisation peut donc être conduite sur 7 paramètres, que nous encapsulons dans le vecteur $\boldsymbol{\delta}$:

$$\boldsymbol{\delta}_k^\mathsf{T} \sim (\mathbf{u}_k^\mathsf{T} \quad \mathbf{v}_k^\mathsf{T} \quad \delta_k).$$

Etudions les propriétés de consistance et de complétude de ce schéma de mise à jour.

4.4.5.1 Consistance

Le schéma de mise à jour (4.19) est consistant si les contraintes de consistance (4.16) et (4.17) de la représentation orthonormale sont garanties sur $(\mathsf{U}_{k+1}, \mathsf{V}_{k+1}, \sigma_{k+1})$. Concernant U, il est trivial de voir que si $\mathsf{U}_k \in O(3)$, comme $\mathsf{R}_{3D}(\mathbf{u}_k) \in SO(3)$, alors $\mathsf{U}_k \mathsf{R}_{3D}(\mathbf{u}_k) = \mathsf{U}_{k+1} \in O(3)$. De même pour V.

Concernant σ, on constate que $\sigma_{k+1} = \sigma_k + \delta_k$ ne satisfait pas systématiquement les contraintes (4.17). Deux solutions peuvent être envisagées : laisser σ violer les contraintes ou le forcer à les respecter. Si σ viole les contraintes, la loi de recomposition (4.15) garantit tout de même que la matrice formée est une matrice fondamentale. Il est donc très vraisemblable qu'aucun problème pratique ne soit posé. Cependant, il y a plusieurs solutions pour forcer σ à respecter les contraintes tout en laissant la matrice fondamentale correspondante invariante. Nous décrivons une solution.

Borne inférieure ($\sigma > 0$). Supposons pour commencer que $\sigma < 0$. Le cas $\sigma = 0$ sera examiné ensuite. Raisonnons sur la loi de recomposition (4.15). On voit que si les signes de σ et par exemple de $\mathbf{u}_2$ sont changés, la matrice fondamentale n'est pas affectée, d'où les opérations suivantes :

$$\text{Si } \sigma < 0 \text{ alors } \sigma \leftarrow -\sigma \text{ et } \mathbf{u}_2 \leftarrow -\mathbf{u}_2.$$

Notons que ces opérations peuvent entraîner la violation de la contrainte sur la borne supérieure (voir ci-dessous). Le cas $\sigma = 0$ entraîne que la matrice fondamentale correspondante est de rang 1, ce qui n'est géométriquement pas possible. Il n'y a pas de solution pour éviter ce cas. La plupart des paramétrisations ne le gère pas [49, 90, 233, 236] (seule la contrainte $\det(\mathsf{F}) = 0$ est utilisée, et non $\text{rang}(\mathsf{F}) = 2$). La convergence d'un estimateur vers une telle configuration est très peu probable.

Borne supérieure ($\sigma \leq 1$). Multiplions la loi de recomposition (4.15) par $1/\sigma$. On obtient :

$$\mathsf{F} \sim \frac{\mathbf{u}_1 \mathbf{v}_1^\mathsf{T}}{\sigma} + \mathbf{u}_2 \mathbf{v}_2^\mathsf{T}.$$

L'idée est d'échanger les colonnes $\mathbf{u}_1$ et $\mathbf{u}_2$ puis $\mathbf{v}_1$ et $\mathbf{v}_2$. On pourra alors remplacer σ par $1/\sigma$, qui satisfera à nouveau la contrainte (4.17), d'où les opérations suivantes :

$$\text{Si } \sigma > 1 \text{ alors } \sigma \leftarrow \frac{1}{\sigma} \text{ ; échanger}(\mathbf{u}_1, \mathbf{u}_2) \text{ et échanger } (\mathbf{v}_1, \mathbf{v}_2).$$

4.4.5.2 Complétude

Le schéma de mise à jour (4.19) est complet si toute matrice fondamentale peut être atteinte à partir de toute initialisation. Concernant U et V la question suivante se pose. Ces matrices orthonormales peuvent avoir

un déterminant positif ou négatif et sont mises à jour par une multiplication qui ne change pas le signe de leur déterminant (puisque $\det(\mathsf{M}) = 1$ pour $\mathsf{M} \in SO(3)$) :

$$\text{signe}(\det(\mathsf{U}_{k+1})) \quad = \quad \text{signe}(\det(\mathsf{U}_k)).$$

Ceci n'est pas un problème car pour toute représentation orthonormale, on peut changer le signe de U et V (ce qui change le signe de leur déterminant : $\det(\mathsf{U}) = -\det(-\mathsf{U})$), sans changer la matrice fondamentale représentée :

$$\mathsf{F} \quad \sim \quad (\pm\mathsf{U})\Sigma(\pm\mathsf{V})^{\mathsf{T}}.$$

Ce sont les ambiguïtés discrètes évoquées précédemment. Concernant σ, il est trivial de voir que le schéma de mise à jour (4.19) est complet.

4.4.5.3 Optimisation

L'application des techniques d'optimisation de moindres carrés non-linéaires nécessite la matrice Jacobienne du vecteur d'erreurs par rapport aux paramètres de mise à jour. Les fonctions d'erreur sont exprimées en termes de la matrice fondamentale, ou en termes des matrices de projection. Nous donnons ci-dessous les matrices Jacobiennes correspondant au schéma de mise à jour (4.19), évaluées à l'estimation courante, c'est à dire pour $\boldsymbol{\delta}_k = \mathbf{0}_{(7\times1)}$, pour la matrice fondamentale et la réalisation canonique singulière :

$$\begin{aligned}(\mathsf{A}_{\mathsf{F}}^{\perp})_{(9\times7)} &= \left.\frac{\partial \mathbf{f}_{k+1}}{\partial \boldsymbol{\delta}_k}\right|_{\mathsf{F}_k} \\ (\mathsf{A}_{\check{\mathsf{P}}'}^{\perp})_{(12\times7)} &= \left.\frac{\partial \check{\mathbf{p}}'_{k+1}}{\partial \boldsymbol{\delta}_k}\right|_{\check{\mathsf{P}}'_k},\end{aligned}$$

où $\mathbf{f} = \text{vect}(\mathsf{F})$ et $\check{\mathbf{p}}' = \text{vect}(\check{\mathsf{P}}')$ sont les vectorisations par ligne respectives de la matrice fondamentale et de la deuxième matrice de projection canonique singulière. Ces matrices Jacobiennes peuvent être ensuite utilisées pour la minimisation de différentes fonctions de coût

La matrice Jacobienne $\mathsf{A}_{\mathsf{F}}^{\perp}$. Nous formulons la matrice $\mathsf{A}_{\mathsf{F}}^{\perp}$. Pour plus de lisibilité, l'évaluation à l'estimation courante ne sera pas notée explicitement :

$$\mathsf{A}_{\mathsf{F}}^{\perp} \quad = \quad \frac{\partial \mathbf{f}_{k+1}}{\partial \boldsymbol{\delta}_k} \quad = \quad \left(\frac{\partial \mathbf{f}_{k+1}}{\partial u_{k,1}} \quad \cdots \quad \frac{\partial \mathbf{f}_{k+1}}{\partial v_{k,3}} \quad \frac{\partial \mathbf{f}_{k+1}}{\partial \delta_k}\right).$$

Examinons la première colonne. Comme $\mathbf{f}_{k+1} = \text{vect}(\mathsf{F}_{k+1})$, elle s'écrit :

$$\frac{\partial \mathbf{f}_{k+1}}{\partial u_{k,1}} \quad = \quad \text{vect}\left(\frac{\partial \mathsf{F}_{k+1}}{\partial u_{k,1}}\right).$$

En remplaçant F_{k+1} par l'expression de recomposition (4.15), puis en évaluant cette expression à $\boldsymbol{\delta}_k = \mathbf{0}_{(7\times1)}$, on obtient :

$$\frac{\partial \mathbf{f}}{\partial u_{k,1}} \quad = \quad \text{vect}\left(\frac{\partial \mathsf{U}_k \mathsf{R}_{3D}(\mathbf{u}_k)\Sigma_{k+1}\mathsf{R}_{3D}(\mathbf{v}_k)^{\mathsf{T}}\mathsf{V}_k^{\mathsf{T}}}{\partial u_{k,1}}\right) \quad = \quad \text{vect}\left(\mathsf{U}_k \frac{\partial \mathsf{R}_{3D}(\mathbf{u}_k)}{\partial u_{k,1}}\Sigma_k \mathsf{V}_k^{\mathsf{T}}\right).$$

D'après l'équation (2.39), nous obtenons :

$$\frac{\partial \mathbf{f}}{\partial u_{k,1}} \quad = \quad \text{vect}\left((\mathbf{0} \;\; \mathbf{u}_{k,3} \;\; -\mathbf{u}_{k,2})\Sigma_k \mathsf{V}_k^{\mathsf{T}}\right) \quad = \quad \text{vect}\left(\sigma_k \mathbf{u}_{k,3}\mathbf{v}_{k,2}^{\mathsf{T}}\right),$$

où les $\mathbf{u}_{k,i}$ et les $\mathbf{v}_{k,i}$ sont les colonnes de U_k et de V_k respectivement. En suivant un raisonnement similaire, nous obtenons les autres colonnes de $\mathsf{A}_{\mathsf{F}}^{\perp}$:

$$(\mathsf{A}_{\mathsf{F}}^{\perp})^{\mathsf{T}} = \begin{pmatrix} \text{vect}\left(\sigma_k \mathbf{u}_{k,3}\mathbf{v}_{k,2}^{\mathsf{T}}\right)^{\mathsf{T}} \\ \text{vect}\left(-\mathbf{u}_{k,3}\mathbf{v}_{k,1}^{\mathsf{T}}\right)^{\mathsf{T}} \\ \text{vect}\left(\mathbf{u}_{k,2}\mathbf{v}_{k,1}^{\mathsf{T}} - \sigma_k \mathbf{u}_{k,1}\mathbf{v}_{k,2}^{\mathsf{T}}\right)^{\mathsf{T}} \\ \text{vect}\left(\sigma_k \mathbf{u}_{k,2}\mathbf{v}_{k,3}^{\mathsf{T}}\right)^{\mathsf{T}} \\ \text{vect}\left(-\mathbf{u}_{k,1}\mathbf{v}_{k,3}^{\mathsf{T}}\right)^{\mathsf{T}} \\ \text{vect}\left(\mathbf{u}_{k,1}\mathbf{v}_{k,2}^{\mathsf{T}} - \sigma_k \mathbf{u}_{k,2}\mathbf{v}_{k,1}^{\mathsf{T}}\right)^{\mathsf{T}} \\ \text{vect}\left(\mathbf{u}_{k,2}\mathbf{v}_{k,2}^{\mathsf{T}}\right)^{\mathsf{T}} \end{pmatrix}, \tag{4.20}$$

ou, de manière explicite (l'indice k est ignoré pour cette expression, u_{ij} et v_{ij} indiquent les coefficients des matrices U_k et V_k, et $\sigma = \sigma_k$) :

$$\mathsf{A}_{\mathsf{F}}^{\perp} = \begin{pmatrix} \sigma u_{13}v_{12} & -u_{13}v_{11} & u_{12}v_{11} - \sigma u_{11}v_{12} & \sigma u_{12}v_{13} & -u_{11}v_{13} & u_{11}v_{12} - \sigma u_{12}v_{11} & u_{12}v_{12} \\ \sigma u_{13}v_{22} & -u_{13}v_{21} & u_{12}v_{21} - \sigma u_{11}v_{22} & \sigma u_{12}v_{23} & -u_{11}v_{23} & u_{11}v_{22} - \sigma u_{12}v_{21} & u_{12}v_{22} \\ \sigma u_{13}v_{32} & -u_{13}v_{31} & u_{12}v_{31} - \sigma u_{11}v_{32} & \sigma u_{12}v_{33} & -u_{11}v_{33} & u_{11}v_{32} - \sigma u_{12}v_{31} & u_{12}v_{32} \\ \sigma u_{23}v_{12} & -u_{23}v_{11} & u_{22}v_{11} - \sigma u_{21}v_{12} & \sigma u_{22}v_{13} & -u_{21}v_{13} & u_{21}v_{12} - \sigma u_{22}v_{11} & u_{22}v_{12} \\ \sigma u_{23}v_{22} & -u_{23}v_{21} & u_{22}v_{21} - \sigma u_{21}v_{22} & \sigma u_{22}v_{23} & -u_{21}v_{23} & u_{21}v_{22} - \sigma u_{22}v_{21} & u_{22}v_{22} \\ \sigma u_{23}v_{32} & -u_{23}v_{31} & u_{22}v_{31} - \sigma u_{21}v_{32} & \sigma u_{22}v_{33} & -u_{21}v_{33} & u_{21}v_{32} - \sigma u_{22}v_{31} & u_{22}v_{32} \\ \sigma u_{33}v_{12} & -u_{33}v_{11} & u_{32}v_{11} - \sigma u_{31}v_{12} & \sigma u_{32}v_{13} & -u_{31}v_{13} & u_{31}v_{12} - \sigma u_{32}v_{11} & u_{32}v_{12} \\ \sigma u_{33}v_{22} & -u_{33}v_{21} & u_{32}v_{21} - \sigma u_{31}v_{22} & \sigma u_{32}v_{23} & -u_{31}v_{23} & u_{31}v_{22} - \sigma u_{32}v_{21} & u_{32}v_{22} \\ \sigma u_{33}v_{32} & -u_{33}v_{31} & u_{32}v_{31} - \sigma u_{31}v_{32} & \sigma u_{32}v_{33} & -u_{31}v_{33} & u_{31}v_{32} - \sigma u_{32}v_{31} & u_{32}v_{32} \end{pmatrix}.$$

La matrice Jacobienne $\mathsf{A}_{\mathsf{P}'}^{\perp}$. La formulation de la matrice Jacobienne $\mathsf{A}_{\mathsf{P}'}^{\perp}$, donnant les variations de la deuxième matrice de projection canonique singulière, équation (2.44), suit un raisonnement similaire à celle de $\mathsf{A}_{\mathsf{F}}^{\perp}$. Nous ne donnons donc que son expression finale :

$$(\mathsf{A}_{\mathsf{P}'}^{\perp})^{\mathsf{T}} = \begin{pmatrix} \text{vect}\left(\begin{pmatrix} \mathbf{u}_{k,3}\mathbf{v}_{k,1}^{\mathsf{T}} & -\mathbf{u}_{k,2} \end{pmatrix}\right)^{\mathsf{T}} \\ \text{vect}\left(\begin{pmatrix} \sigma_k \mathbf{u}_{k,3}\mathbf{v}_{k,2}^{\mathsf{T}} & -\mathbf{u}_{k,1} \end{pmatrix}\right)^{\mathsf{T}} \\ \text{vect}\left(\begin{pmatrix} -\mathbf{u}_{k,1}\mathbf{v}_{k,1}^{\mathsf{T}} - \sigma_k \mathbf{u}_{k,2}\mathbf{v}_{k,2}^{\mathsf{T}} & \mathbf{0} \end{pmatrix}\right)^{\mathsf{T}} \\ \text{vect}\left(\begin{pmatrix} -\sigma_k \mathbf{u}_{k,1}\mathbf{v}_{k,3}^{\mathsf{T}} & \mathbf{0} \end{pmatrix}\right)^{\mathsf{T}} \\ \text{vect}\left(\begin{pmatrix} -\mathbf{u}_{k,2}\mathbf{v}_{k,3}^{\mathsf{T}} & \mathbf{0} \end{pmatrix}\right)^{\mathsf{T}} \\ \text{vect}\left(\begin{pmatrix} -\mathbf{u}_{k,2}\mathbf{v}_{k,2}^{\mathsf{T}} - \sigma_k \mathbf{u}_{k,1}\mathbf{v}_{k,1}^{\mathsf{T}} & \mathbf{0} \end{pmatrix}\right)^{\mathsf{T}} \\ \text{vect}\left(\begin{pmatrix} -\mathbf{u}_{k,1}\mathbf{v}_{k,2}^{\mathsf{T}} & \mathbf{0} \end{pmatrix}\right)^{\mathsf{T}} \end{pmatrix}, \tag{4.21}$$

et de manière explicite (avec les mêmes notations que pour $\mathsf{A}_{\mathsf{F}}^{\perp}$) :

$$\mathsf{A}_{\check{\mathsf{P}}'}^{\perp} = \begin{pmatrix}
u_{13}v_{11} & \sigma u_{13}v_{12} & -u_{11}v_{11}-\sigma u_{12}v_{12} & -\sigma u_{11}v_{13} & -u_{12}v_{13} & -u_{12}v_{12}-\sigma u_{11}v_{11} & -u_{11}v_{12} \\
u_{13}v_{21} & \sigma u_{13}v_{22} & -u_{11}v_{21}-\sigma u_{12}v_{22} & -\sigma u_{11}v_{23} & -u_{12}v_{23} & -u_{12}v_{12}-\sigma u_{11}v_{21} & -u_{11}v_{22} \\
u_{13}v_{31} & \sigma u_{13}v_{22} & -u_{11}v_{31}-\sigma u_{12}v_{32} & -\sigma u_{11}v_{33} & -u_{12}v_{33} & -u_{12}v_{12}-\sigma u_{11}v_{31} & -u_{11}v_{32} \\
-u_{12} & u_{11} & 0 & 0 & 0 & 0 & 0 \\
u_{23}v_{11} & \sigma u_{23}v_{12} & -u_{21}v_{11}-\sigma u_{22}v_{12} & -\sigma u_{21}v_{13} & -u_{22}v_{13} & -u_{22}v_{12}-\sigma u_{21}v_{11} & -u_{21}v_{12} \\
u_{23}v_{21} & \sigma u_{23}v_{22} & -u_{21}v_{21}-\sigma u_{22}v_{22} & -\sigma u_{21}v_{23} & -u_{22}v_{23} & -u_{22}v_{12}-\sigma u_{21}v_{21} & -u_{21}v_{22} \\
u_{23}v_{31} & \sigma u_{23}v_{22} & -u_{21}v_{31}-\sigma u_{22}v_{32} & -\sigma u_{21}v_{33} & -u_{22}v_{33} & -u_{22}v_{12}-\sigma u_{21}v_{31} & -u_{21}v_{32} \\
-u_{22} & u_{21} & 0 & 0 & 0 & 0 & 0 \\
u_{33}v_{11} & \sigma u_{33}v_{12} & -u_{31}v_{11}-\sigma u_{32}v_{12} & -\sigma u_{31}v_{13} & -u_{32}v_{13} & -u_{32}v_{12}-\sigma u_{31}v_{11} & -u_{31}v_{12} \\
u_{33}v_{21} & \sigma u_{33}v_{22} & -u_{31}v_{21}-\sigma u_{32}v_{22} & -\sigma u_{31}v_{23} & -u_{32}v_{23} & -u_{32}v_{12}-\sigma u_{31}v_{21} & -u_{31}v_{22} \\
u_{33}v_{31} & \sigma u_{33}v_{22} & -u_{31}v_{31}-\sigma u_{32}v_{32} & -\sigma u_{31}v_{33} & -u_{32}v_{33} & -u_{32}v_{12}-\sigma u_{31}v_{31} & -u_{31}v_{32} \\
-u_{32} & u_{31} & 0 & 0 & 0 & 0 & 0
\end{pmatrix}.$$

Décomposition : de la matrice fondamentale F à la représentation orthonormale $(\mathsf{U}, \mathsf{V}, \sigma)$:
▷ calculer la décomposition en valeurs singulières $\mathsf{F} = \mathsf{U}\,\mathrm{diag}(\sigma_1, \sigma_2, 0)\mathsf{V}^{\mathsf{T}}$;
▷ affecter $\sigma = \sigma_2/\sigma_1$.
Recomposition : de la représentation orthonormale $(\mathsf{U}, \mathsf{V}, \sigma)$ à la matrice fondamentale F ou à la deuxième matrice de projection canonique singulière $\check{\mathsf{P}}'$:
▷ $\mathsf{F} \sim \mathbf{u}_1\mathbf{v}_1^{\mathsf{T}} + \sigma\mathbf{u}_2\mathbf{v}_2^{\mathsf{T}}$;
▷ $\check{\mathsf{P}}' \sim (\ \mathbf{u}_2\mathbf{v}_1^{\mathsf{T}} - \sigma\mathbf{u}_1\mathbf{v}_2^{\mathsf{T}} \quad \mathbf{u}_3)$.
Mise à jour locale minimale (k est le compteur d'itérations) :
▷ *Paramètres d'optimisation* : $\boldsymbol{\delta}_k^{\mathsf{T}} = (\underbrace{u_{k,1}\ u_{k,2}\ u_{k,3}}_{\mathbf{u}_k^{\mathsf{T}}}\ \underbrace{v_{k,1}\ v_{k,2}\ v_{k,3}}_{\mathbf{v}_k^{\mathsf{T}}}\ \delta_k)$;
▷ *Loi de mise à jour* (voir le chapitre 2 §2.7.3) :
- $\mathsf{U}_{k+1} = \mathsf{U}_k\mathsf{R}_{3D}(\mathbf{u}_k)$,
- $\mathsf{V}_{k+1} = \mathsf{V}_k\mathsf{R}_{3D}(\mathbf{v}_k)$,
- $\sigma_{k+1} = \sigma_k + \delta_k$;

▷ *Matrices Jacobiennes associées* : voir équations (4.20) et (4.21) pour la matrice fondamentale et la deuxième matrice de projection canonique singulière respectivement.

TAB. 4.1 – La *représentation orthonormale de la matrice fondamentale*. Ce tableau résume les formules de lien avec la matrice fondamentale, et donne le schéma de mise à jour locale minimale de la représentation orthonormale ainsi que les matrices Jacobiennes $\mathsf{A}_{\mathsf{F}}^{\perp}$ et $\mathsf{A}_{\check{\mathsf{P}}'}^{\perp}$ exprimant respectivement les variations de la matrice fondamentale et de la deuxième matrice de projection canonique singulière par rapport aux paramètres de mise à jour.

4.4.5.4 Estimation partielle

La géométrie épipolaire se décompose en une paire d'épipôles et une transformation épipolaire. Si une ou deux de ces entités est connue a priori, il est pratique de pouvoir n'estimer que celle(s) restant inconnue(s). Lorsqu'une paramétrisation exprimée directement en termes de ces entités est utilisée, voir §4.3, l'incorporation de ces informations est directe. Lorsqu'une paramétrisation basée sur les matrices de projection est utilisée, cela devient plus difficile. Lorsque la représentation orthonormale est utilisée, la solution simple suivante peut être adoptée.

Fixer un épipôle. Considérons par exemple l'épipôle $\boldsymbol{e}$, encapsulé dans la représentation orthonormale par la $3^{\text{ème}}$ colonne de V. La mise à jour par le schéma (4.19) ne change pas cet épipôle si $v_1 = v_2 = 0$. En

conséquence, on peut laisser la position de cet épipôle invariante en enlevant v_1 et v_2 de l'optimisation et en mettant à jour la matrice V par $\mathsf{V}_{k+1} = \mathsf{V}_k\mathsf{R}_z(v_3)$. Similairement, on peut fixer l'épipôle $\boldsymbol{e'}$ par la mise à jour suivante de la matrice U : $\mathsf{U}_{k+1} = \mathsf{U}_k\mathsf{R}_z(u_3)$.

Fixer la transformation épipolaire. Les épipôles sont encapsulés par les paramètres de mise à jour u_1, u_2 et v_1, v_2. Par conséquent les 3 degrés de liberté de la transformation épipolaire sont contenus dans les paramètres de mise à jour suivants : u_3, v_3 et σ. Pour laisser la transformation épipolaire invariante, il suffit d'enlever ces paramètres de l'optimisation.

4.4.6 L'approche réduite : minimisation des distances point-droite épipolaire

Nous étudions l'utilisation de la représentation orthonormale pour la minimisation des distances entre les points et les droites épipolaires correspondantes. C'est la fonction de coût de la méthode réduite [218].

4.4.6.1 Fonction de coût

Soient $\boldsymbol{q}_j \leftrightarrow \boldsymbol{q}'_j$, $j = 1 \dots m$, m correspondances de points. La fonction de coût est donnée par :

$$\mathcal{C}(\mathsf{F}) \quad = \quad \sum_j \left(d^2_{PH}(\mathbf{q}_j, \mathsf{F}^\mathsf{T}\mathbf{q}'_j) + d^2_{PH}(\mathbf{q}'_j, \mathsf{F}\mathbf{q}_j) \right), \tag{4.22}$$

où d^2_{PH} est la distance orthogonale (Euclidienne) entre un point et une droite du plan définie par l'équation (2.14). La fonction de coût (4.22) est une somme de carrés non-linéaires en les coefficients de F. Définissons le vecteur d'erreurs résiduelles $\mathbf{r}$ de taille m par :

$$\mathbf{r} = \begin{pmatrix} \vdots \\ \pm\sqrt{d^2_{PH}(\mathbf{q}_j, \mathsf{F}^\mathsf{T}\mathbf{q}'_j) + d^2_{PH}(\mathbf{q}'_j, \mathsf{F}\mathbf{q}_j)} \\ \vdots \end{pmatrix}.$$

La fonction de coût (4.22) se formule alors comme :

$$\mathcal{C}(\mathsf{F}) \quad = \quad \mathbf{r}^\mathsf{T}\mathbf{r} \quad = \quad \|\mathbf{r}\|^2,$$

et le problème à résoudre est un problème d'optimisation de moindres carrés non-linéaires. Nous utilisons les techniques présentées au chapitre 3. La matrice Jacobienne suivante est nécessaire :

$$\mathsf{J} \quad = \quad \frac{\partial \mathbf{r}}{\partial \boldsymbol{\delta}} \quad = \quad \mathsf{A}\mathsf{A}_\mathsf{F}^\perp \ .$$

Comme expliqué en §3.5.2, il est pratique d'exprimer cette matrice comme un produit de deux matrices. Soit $\mathbf{f} = \text{vect}(\mathsf{F})$ la vectorisation par ligne de la matrice F. La matrice $\mathsf{A} = \frac{\partial \mathbf{r}}{\partial \mathbf{f}}$ est la matrice Jacobienne de la fonction de coût et dépend uniquement de cette dernière. Une formule pour le calcul de A est donnée ci-dessous. L'autre matrice, $\mathsf{A}_\mathsf{F}^\perp = \frac{\partial \mathbf{f}}{\partial \boldsymbol{\delta}}$, dépend uniquement de la paramétrisation et de la loi de mise à jour employées, ici la représentation orthonormale. Elle est donnée par l'équation (4.20).

4.4.6.2 Matrice Jacobienne de la fonction de coût

Nous désirons former la matrice Jacobienne $\mathsf{A} = \frac{\partial \mathbf{r}}{\partial \mathbf{f}}$. Dans cette section, l'indice j n'est pas écrit explicitement. On utilise k comme indice du vecteur $\mathbf{f}$ et désignons par F_{rc} le coefficient correspondant dans la matrice F. On note les lignes de F comme $\mathbf{r}_1$, $\mathbf{r}_2$ et $\mathbf{r}_3$ et ses colonnes comme $\mathbf{c}_1$, $\mathbf{c}_2$ et $\mathbf{c}_3$. On définit $\alpha_r = \mathbf{r}_r^\mathsf{T}\mathbf{q}$,

$r \in \{1,2\}$, $\alpha_3 = 0$ et $\beta_c = \mathbf{c}_c^\mathsf{T}\mathbf{q}'$, $c \in \{1,2\}$, $\beta_3 = 0$. En utilisant la définition (2.14) de la distance Euclidienne point-droite, développons chaque terme e du vecteur d'erreurs résiduelles :

$$\begin{aligned} e &= \pm\sqrt{d_{PH}^2(\mathbf{q}_j, \mathsf{F}^\mathsf{T}\mathbf{q}'_j) + d_{PH}^2(\mathbf{q}'_j, \mathsf{F}\mathbf{q}_j)} \\ &= \underbrace{\sqrt{\frac{1}{\alpha_1^2+\alpha_2^2} + \frac{1}{\beta_1^2+\beta_2^2}}}_{\nu} \underbrace{\mathbf{q}'^\mathsf{T}\mathsf{F}\mathbf{q}}_{\xi}. \end{aligned}$$

Après développement, en particulier en utilisant les formules, $\frac{\partial(\alpha_1^2+\alpha_2^2)}{\partial f_k} = 2\alpha_i q_j$, $\frac{\partial(\beta_1^2+\beta_2^2)}{\partial f_k} = 2\beta_j q'_i$ et $\frac{\partial \xi}{\partial f_k} = q_j q'_i$, on obtient :

$$\frac{\partial e}{\partial f_k} = \nu q_j q'_i - \left(\frac{\alpha_i q_j}{\nu(\alpha_1^2+\alpha_2^2)^2} + \frac{\beta_j q'_i}{\nu(\beta_1^2+\beta_2^2)^2} \right).$$

4.4.7 L'approche directe : minimisation de l'erreur de reprojection

Nous étudions l'utilisation de la représentation orthonormale pour la minimisation de l'erreur de reprojection, c'est à dire pour l'ajustement de faisceaux projectif avec deux vues. La fonction de coût est donnée par l'équation (3.1). La reconstruction est exprimée dans une base projective déterminée par les matrices de projection canoniques. La deuxième matrice de projection canonique est extraite de la représentation orthonormale via l'équation (4.18). La résolution éparse des équations normales sera bien sûr conduite à l'avantage des paramètres des points, quasiment toujours plus nombreux que ceux des caméras lorsque seulement deux vues sont considérées. Nous insérons notre analyse dans le cadre global donné dans [93]. Le tableau 4.2 résume les modifications.

Ajustement de faisceaux projectif pour deux vues exprimé dans le cadre donné dans [93] (algorithme A4.1). L'estimation initiale de la matrice fondamentale est F_0.

Ajouter les étapes suivantes :

(i') initialiser la représentation orthonormale $(\mathsf{U}, \mathsf{V}, \sigma)$ par une décomposition en valeurs singulières : $\boxed{\mathsf{F}_0 \sim \mathsf{U}\ \mathrm{diag}(1,\sigma,0)\ \mathsf{V}^\mathsf{T}}$;

(ii') changer la matrice Jacobienne $(r \times 12)$ A en une matrice Jacobienne minimale $(r \times 7)$ pour la représentation orthonormale : $\boxed{\mathsf{A} \leftarrow \mathsf{A}\,\mathsf{A}_{\hat{\mathsf{P}}'}^{\perp}}$, où $\mathsf{A}_{\hat{\mathsf{P}}'}^{\perp}$ est donnée par l'équation (4.21).

Changer l'étape de mise à jour des paramètres en :

(viii) mise à jour de la représentation orthonormale :

$$\boxed{\mathsf{U} \leftarrow \mathsf{U}\mathsf{R}_{3D}(\mathbf{u})} \qquad \boxed{\mathsf{V} \leftarrow \mathsf{V}\mathsf{R}_{3D}(\mathbf{v})} \qquad \boxed{\sigma \leftarrow \sigma + \delta},$$

où $\boldsymbol{\delta}^\mathsf{T} = (\mathbf{u}^\mathsf{T}\ \mathbf{v}^\mathsf{T}\ \delta)$ sont les 7 paramètres de mise à jour de la représentation orthonormale. La mise à jour de la structure ne change pas.

TAB. 4.2 – Implantation de notre estimateur minimal basé sur la représentation orthonormale dans le cadre de l'ajustement de faisceaux donnée en [93] (algorithme A4.1). Notons que r désigne le nombre de résidus et que la deuxième matrice de projection doit être extraite de la représentation orthonormale par l'équation (4.18), par exemple pour le calcul des résidus.

4.4.8 Matrices essentielles

Comme remarqué précédemment, lorsque la matrice fondamentale considérée est une matrice essentielle, la représentation orthonormale n'est pas unique. Elle a 2 degrés d'ambiguïté car les matrices essentielles constituent un sous-espace des matrices fondamentales de dimension 5.

Une telle configuration est rencontrée lorsque l'origine du repère image coïncide avec le point principal. En pratique un tel cas peut arriver lorsque des coordonnées calibrées ou semi-calibrées sont utilisées. De même, la technique de normalisation des points image proposée par Hartley [82] peut avoir pour effet de rapprocher la configuration vers une configuration semi-calibrée. Cette normalisation est basée entre autres sur le déplacement de l'origine du repère image sur le centre de gravité des points observés. Si les points observés sont bien répartis dans l'image, leur centre de gravité est proche du centre de l'image, et donc du point principal. Ceci tend à créer des configurations proches d'une semi-calibration, où la matrice fondamentale à estimer sera proche d'une matrice essentielle.

La conséquence d'une telle configuration est que la représentation orthonormale n'est plus unique : si $(\mathsf{U}, \mathsf{V}, \sigma = 1)$ est une représentation orthonormale d'une matrice essentielle, alors on constate sur l'équation de recomposition (4.15) que $\forall \alpha \in \mathbb{R}$, $(\mathsf{U}\mathsf{R}_z(\alpha), \mathsf{V}\mathsf{R}_z(\alpha), \sigma = 1)$ est une autre représentation orthonormale de la même matrice essentielle. Une conséquence gênante est que les matrices Jacobiennes (4.20) et (4.21) sont singulières, ou mal conditionnées. Le processus d'optimisation doit prendre cela en compte, pour éviter les situations singulières.

Nous proposons deux solutions pour gérer ce problème. La première solution consiste à utiliser une technique d'optimisation non-linéaire indépendante du rang de la matrice Jacobienne. Dans la section expérimentale, nous montrons qu'avec l'algorithme de Levenberg-Marquardt, les situations singulières n'induisent pas d'instabilité numérique. De manière plus générale, les méthodes de moindres carrés non-linéaires incluant une étape de régularisation gèrent ce genre de cas, voir §3.6.

La deuxième solution consiste à éviter les situations singulières en utilisant une normalisation des points image appropriée. L'idée est de déplacer l'origine du repère image pour qu'il ne se situe pas près du centre de l'image, ou de s'assurer que σ est proche d'une valeur optimale σ_{opt}, selon un certain critère. Cette valeur peut être déterminée, par exemple, en maximisant le rang des matrices Jacobiennes $\mathsf{A}_{\tilde{\mathsf{P}}'}^{\perp}$ ou $\mathsf{A}_{\mathsf{F}}^{\perp}$. En particulier, on montre que $\sigma_{\text{opt}} = \frac{1}{\sqrt{2}} \approx 0,7071$ maximise la fonction $\det(\mathsf{A}_{\mathsf{F}}^{\perp\mathsf{T}}\mathsf{A}_{\mathsf{F}}^{\perp})$, et que $\sigma_{\text{opt}} = \frac{\sqrt{70}+2}{16} \approx 0,6308$ maximise la fonction $\det(\mathsf{A}_{\tilde{\mathsf{P}}'}^{\perp\mathsf{T}}\mathsf{A}_{\tilde{\mathsf{P}}'}^{\perp})$. Ces résultats sont obtenus en formant analytiquement les matrices $\mathsf{A}_{\mathsf{F}}^{\perp\mathsf{T}}\mathsf{A}_{\mathsf{F}}^{\perp}$ et $\mathsf{A}_{\tilde{\mathsf{P}}'}^{\perp\mathsf{T}}\mathsf{A}_{\tilde{\mathsf{P}}'}^{\perp}$. Ces matrices sont de formes simples, dues à l'orthonormalité de U et V : elles sont symétriques diagonales avec un élément (et son symétrique) hors de la diagonale. Nous donnons ci-dessous la matrice $\mathsf{A}_{\mathsf{F}}^{\perp\mathsf{T}}\mathsf{A}_{\mathsf{F}}^{\perp}$:

$$\left(\mathsf{A}_{\mathsf{F}}^{\perp\mathsf{T}}\mathsf{A}_{\mathsf{F}}^{\perp}\right)_{(7\times7)} = \begin{pmatrix} \sigma^2 & & & & & & \\ & 1 & & & & & \\ & & 1+\sigma^2 & & & -2\sigma & \\ & & & \sigma^2 & & & \\ & & & & 1 & & \\ & & -2\sigma & & & 1+\sigma^2 & \\ & & & & & & 1 \end{pmatrix}.$$

Un court calcul montre que le déterminant de cette matrice est donné par :

$$\det(\mathsf{A}_{\mathsf{F}}^{\perp\mathsf{T}}\mathsf{A}_{\mathsf{F}}^{\perp}) = \sigma^8 - 2\sigma^6 + \sigma^4.$$

Nous en recherchons les extrema en annulant sa dérivée par rapport à σ :

$$\begin{aligned} \frac{\partial \det(\mathsf{A}_{\mathsf{F}}^{\perp\mathsf{T}}\mathsf{A}_{\mathsf{F}}^{\perp})}{\partial\sigma} &= 0 \\ 8\sigma^7 - 12\sigma^5 + 4\sigma^3 &= 0, \end{aligned}$$

la solution $\sigma = 0$ n'étant pas admissible, nous simplifions cette équation en :

$$8\sigma^4 - 12\sigma^2 + 4 \quad = \quad 0.$$

En posant $\lambda = \sigma^2$, puis en factorisant, on obtient :

$$(\lambda - 1)(8\lambda - 4) \quad = \quad 0,$$

et les solutions $\sigma_{\text{opt}} = \pm\frac{1}{\sqrt{2}}$ (les solutions ± 1 que l'on remarque dans l'équation ci-dessus correspondent à des minima de $\det(\mathsf{A}_{\mathsf{F}}^{\perp\mathsf{T}}\mathsf{A}_{\mathsf{F}}^{\perp})$).

4.5 Résultats expérimentaux

Nous comparons un algorithme d'ajustement de faisceaux basé sur la représentation orthonormale avec d'autres algorithmes. Nous utilisons des données simulées puis réelles. Nous détaillons ci-dessous les méthodes comparées, les quantités mesurées, la paramétrisation de la structure et le calcul d'une solution initiale.

Méthodes comparées. Nous comparons les paramétrisations suivantes pour la géométrie des deux caméras :

- ▷ LIBRE est une optimisation directe des 24 coefficients de deux matrices de projection. La base projective de reconstruction est laissée libre. Les $24-7 = 17$ paramètres supplémentaires sont les facteurs d'échelle de chaque matrice de projection et les 15 paramètres de la base projective ;
- ▷ NORMALISÉE [140] est similaire à LIBRE, mais fixe la base projective de reconstruction. Pour ce faire, la reconstruction est renormalisée après chaque itération et des contraintes sont incluses au premier ordre dans la minimisation. La base projective de reconstruction, ainsi que les facteurs d'échelle des matrices de projection sont contraintes ;
- ▷ PAR_LIBRE [90] et §4.3.5 est une optimisation directe des 12 coefficients de la deuxième matrice de projection. La base projective de reconstruction est laissée partiellement libre. Les $12-7 = 5$ paramètres supplémentaires sont le facteur d'échelle de la deuxième matrice de projection, l'échelle globale de la scène reconstruite et la position du plan à l'infini de la reconstruction projective ;
- ▷ CARTES [22, 49, 233] utilise une paramétrisation minimale de la matrice fondamentale basée sur plusieurs cartes. Plus de détails sont donnés en §4.3 ;
- ▷ ORTHO utilise la paramétrisation orthonormale proposée dans ce chapitre, voir §4.4.

Quantités mesurées. Nous mesurons deux quantités caractéristiques d'un processus d'ajustement de faisceaux : le temps machine jusqu'à convergence et l'erreur de reprojection à la convergence. Le premier permet de juger le coût de la méthode, et le deuxième permet d'estimer sa précision. Nous arrêtons l'algorithme lorsque la différence entre deux itérations est inférieure à un seuil prédéfini, typiquement compris entre 10^{-4} et 10^{-8}.

Paramétrisation de la structure. Nous utilisons la paramétrisation proposée dans [93], qui consiste à fixer l'échelle des coordonnées homogènes pour que leur $3^{\text{ème}}$ coefficient soit égal à 1. Les 3 autres coefficients libres sont estimés. Cette paramétrisation n'est utilisable que lorsqu'une base canonique avec $\mathsf{P} \sim (\ \mathrm{I} \quad \mathbf{0})$ est utilisée. En conséquence, les méthodes LIBRE et NORMALISÉE ont leur propre paramétrisation de la structure : elles estiment les 4 coefficients des coordonnées homogènes de chaque point.

Calcul d'une solution initiale. Nous calculons une solution initiale pour la matrice fondamentale par l'algorithme des huit points normalisé [82]. Les points sont ensuite reconstruits dans la base canonique singulière par minimisation de leur erreur de reprojection individuelle.

4.5.1 Données simulées

Scène simulée. Nous simulons un nuage de points dans un cube dont la taille du côté est 1 mètre. Deux caméras observent ces points. La configuration standard est la suivante : la distance focale des caméras est 1000 (en pixels) et elles sont situées à 10 mètres du centre du cube. La distance entre les caméras (“baseline” en Anglais) est 1 mètre. Le nombre de points simulés est 50. Nous ajoutons un bruit Gaussien aux positions exactes des points image, avec un écart-type de 2 pixels. Chaque paramètre de cette scène est varié indépendamment pour pouvoir comparer les différentes méthodes dans différentes situations. Les résultats sont des moyennes sur 100 essais. Le calcul des médianes donne des résultats similaires.

Résultats. Les figures 4.4, 4.5, 4.6 et 4.7 montrent les résultats obtenus, lorsque différents paramètres d’observation sont modifiés. Nous commentons ces résultats de manière globale puis spécifiquement.

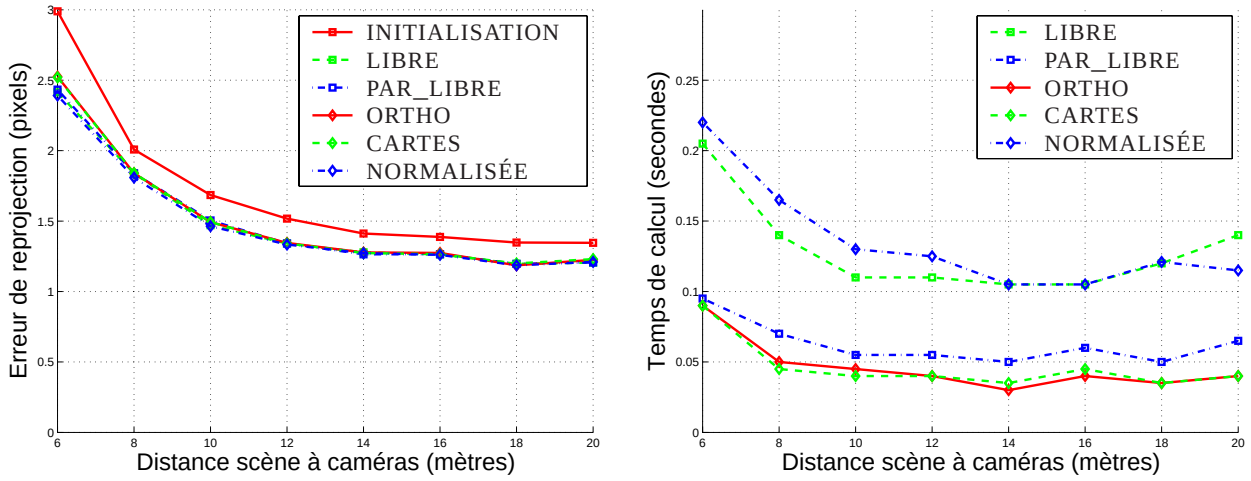


FIG. 4.4 – Erreur de reprojection (à gauche) et temps de calcul (à droite) mesurés pour différentes distances scène à caméras.

Nous observons que dans tous les cas, toutes les méthodes donnent la même erreur de reprojection, à quelques variations près. Ces petites variations sont dues au fait que les différentes paramétrisations sont plus ou moins susceptibles de tomber dans des minima locaux.

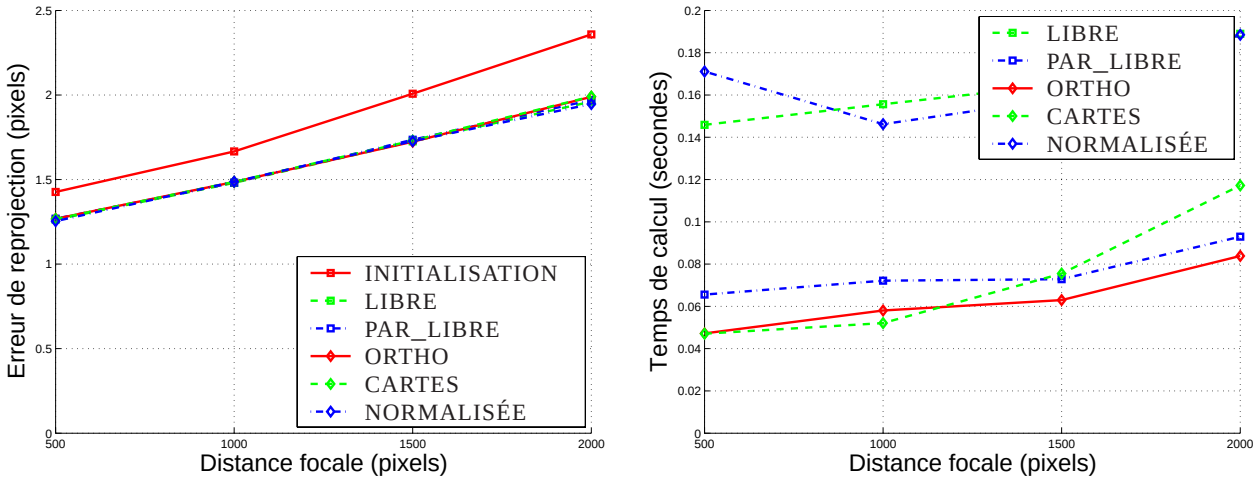


FIG. 4.5 – Erreur de reprojection (à gauche) et temps de calcul (à droite) mesurés pour différentes distances focales des caméras.

D'un autre côté, il y a de grandes différences entre les méthodes, concernant le temps de calcul. Les méthodes ayant les temps de calcul les plus élevés sont NORMALISÉE et LIBRE, suivies par PAR_LIBRE. Les méthodes basées sur un nombre minimal de paramètres, CARTES et ORTHO ont les temps de calcul les plus faibles, à peu près les mêmes. Ces différences sont expliquées par le fait que les méthodes redondantes ont plus d'inconnues à estimer que les méthodes minimales. La résolution des équations normales est donc plus coûteuse. Nous avons observé que dans notre implantation en C, le temps de calcul de chaque itération est dominé par la résolution de ces équations normales, dont la taille est directement liée au nombre de paramètres.

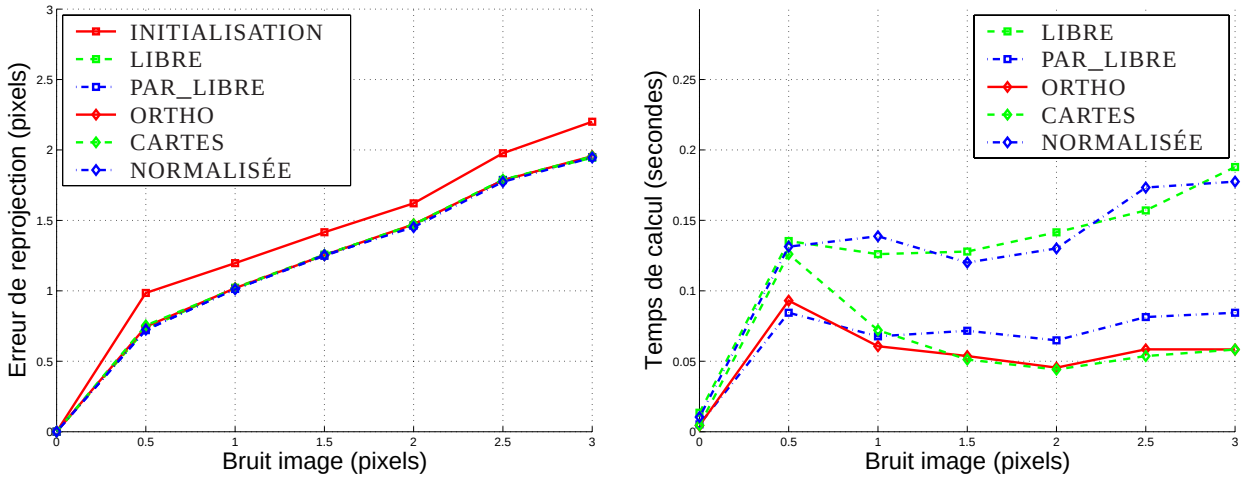


FIG. 4.6 – Erreur de reprojection (à gauche) et temps de calcul (à droite) mesurés pour différents niveaux de bruit ajouté sur les points image.

Plus précisément, la résolution éparse des équations normales est choisie, dans le cas de deux vues, pour être de complexité linéaire en le nombre de points, mais est alors cubique en le nombre de paramètres des caméras, ce qui explique les différences significatives de temps de calcul observées ci-dessus.

On note sur la figure 4.4, que l'erreur diminue lorsque la distance scène à caméras augmente. Ceci est dû au fait que le nuage de points observé dans les images devient de plus en plus petit. La minimisation de l'erreur de reprojection est donc de plus en plus facile.

Ces expériences ont été réalisées avec une régularisation des équations normales basée sur la méthode LM, voir §3.6.2. L'utilisation de la méthode SEBER donne des résultats similaires, mais les temps de calculs sont légèrement moins bons et les différences entre les erreurs de reprojection sont réduites.

Matrices essentielles. Comme remarqué en §4.4.8, la représentation orthonormale a une ambiguïté de dimension 2 quand une matrice essentielle est paramétrée. Nous voulons vérifier si, dans ce cas, ou dans le cas où la configuration considérée est proche d'une matrice essentielle, la représentation orthonormale pourrait induire des instabilités numériques dans le processus d'optimisation. Pour ce faire, nous répétons les expérimentations précédentes, avec les deux changements suivants :

- ▷ nous utilisons comme initialisation la matrice essentielle la plus proche de la matrice fondamentale donnée par l'algorithme des huit points. Par conséquent, la solution visée est une matrice fondamentale, mais la solution initiale est une matrice essentielle ;
- ▷ au lieu d'utiliser les coordonnées des points dans les images, nous utilisons leurs coordonnées sur la rétine. Par conséquent, la géométrie épipolaire sous-jacente est représentée par une matrice essentielle.

Nous avons expérimenté les stratégies LM et SEBER pour l'optimisation non-linéaire. Les résultats obtenus sont très similaires aux expérimentations précédentes. Cela signifie que la représentation orthonormale peut être utilisée pour des configurations géométriques proches d'une matrice essentielle, sans induire de problème au niveau de l'optimisation, lorsqu'une technique d'optimisation appropriée telle Levenberg-Marquardt est utilisée.

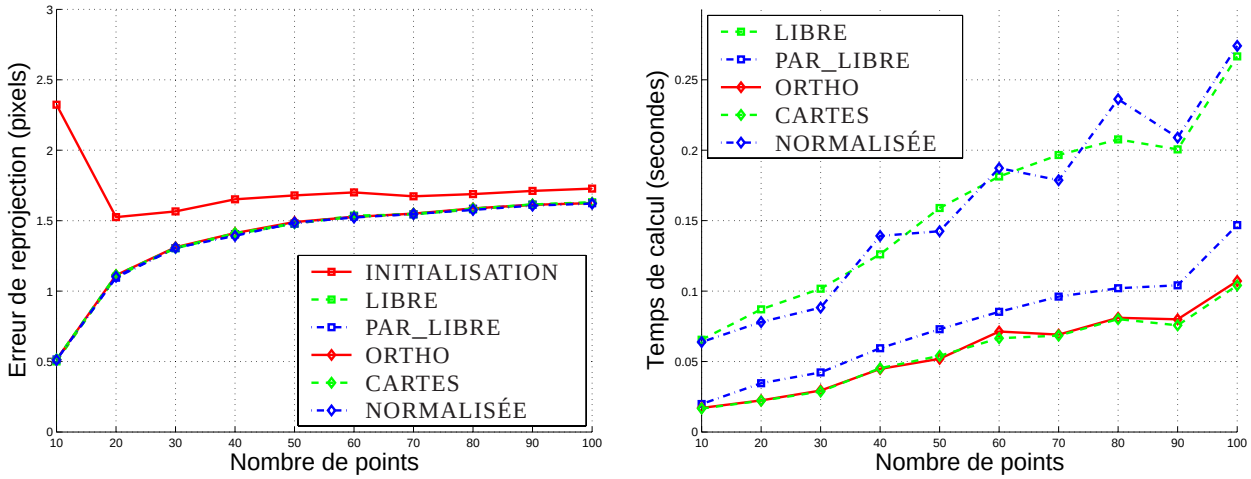


FIG. 4.7 – Erreur de reprojection (à gauche) et temps de calcul (à droite) mesurés pour différents nombres de points.

épipôles		images	INITIALISATION		LIBRE		PAR_LIBRE		ORTHO		CARTES		NORMALISÉE	
e	**e′**		$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$	$\mathcal{E}$	$\mathcal{T}$
~~∞~~	∞	$\mathcal{A}, \mathcal{B}$	0,49	-	0,47	0,99	0,47	0,54	0,47	0,64	0,47	0,65	0,47	1,10
		$\mathcal{A}, \mathcal{C}$	0,68	-	0,65	0,67	0,65	0,39	0,65	0,29	0,65	0,34	0,65	0,70
~~∞~~	~~∞~~	$\mathcal{A}, \mathcal{D}$	0,84	-	0,67	0,70	0,67	0,38	0,67	0,33	0,67	0,33	0,67	0,74
∞	∞	$\mathcal{B}, \mathcal{C}$	0,57	-	0,53	0,27	0,53	0,14	0,53	0,15	0,53	0,14	0,53	0,23
∞	~~∞~~	$\mathcal{B}, \mathcal{D}$	0,79	-	0,55	0,25	0,55	0,23	0,55	0,18	0,55	0,19	0,55	0,25
		$\mathcal{C}, \mathcal{B}$	0,57	-	0,53	0,30	0,53	0,10	0,53	0,12	0,53	0,21	0,53	0,20
moyennes $\mathcal{E}$ et $\mathcal{T}$			0,66	-	0,57	0,53	0,57	0,30	0,57	0,28	0,57	0,31	0,57	0,54

TAB. 4.3 – Erreur de reprojection à convergence, $\mathcal{E}$, et temps de calcul, $\mathcal{T}$, obtenus lorsque des paires d'images de la figure 4.8 sont combinées, pour obtenir des épipôles proche des images ou à l'infini.

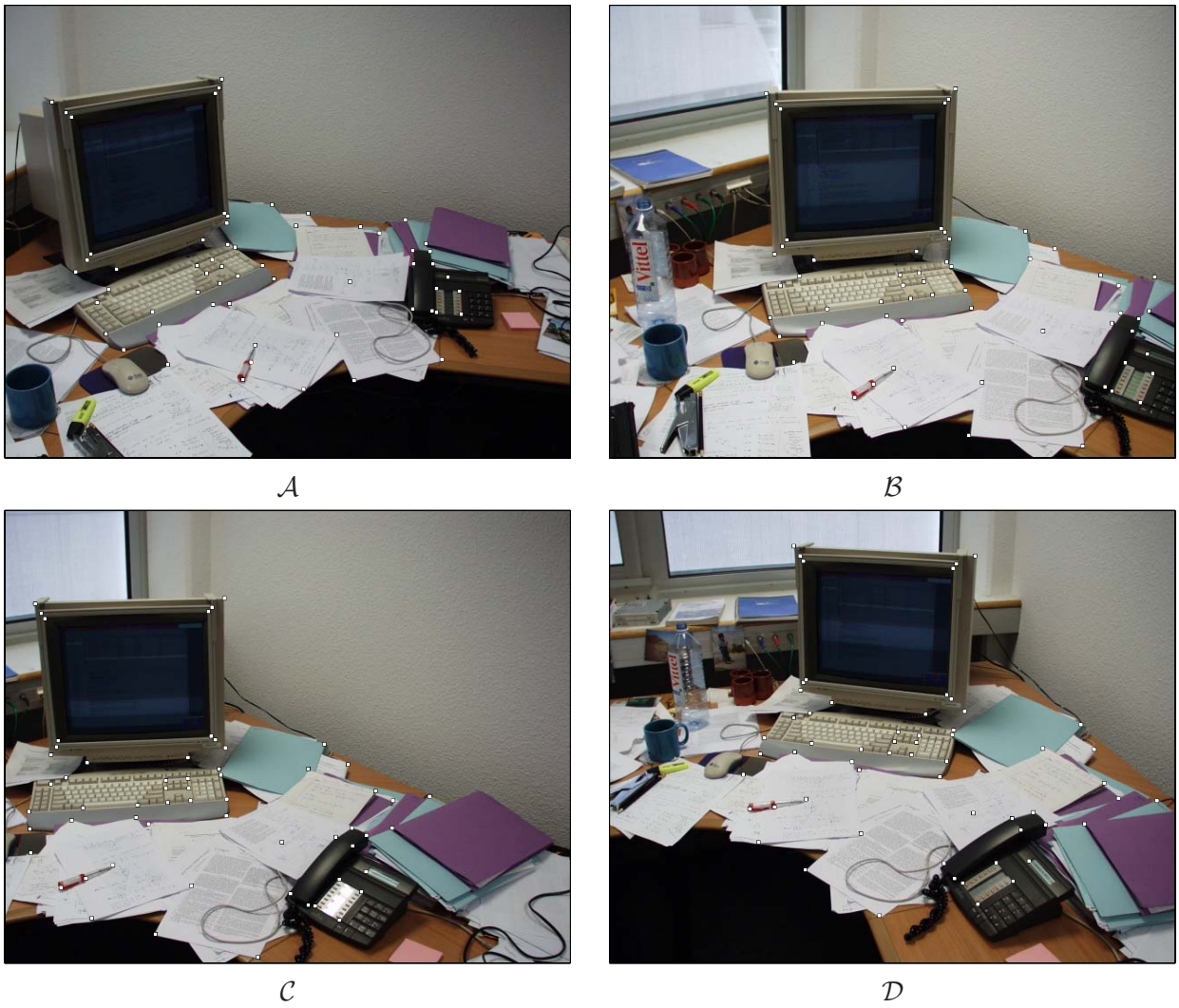
4.5.2 Données réelles

Nous utilisons les images de la figure 4.8. L'idée est de former des paires d'images permettant d'obtenir tous les cas d'épipôles finis et infinis. Les résultats sont donnés dans le tableau 4.3. Pour chaque combinaison de deux images et chaque algorithme, nous calculons le coût et l'erreur de reprojection. La dernière ligne du tableau montre les valeurs moyennes obtenues pour chaque algorithme sur toutes les paires d'images. On remarque que pour chaque paire d'images, tous les algorithmes donnent la même erreur de reprojection. Les méthodes ORTHO, PAR_LIBRE et CARTES donnent les coûts les plus bas, pratiquement deux fois moindres que ceux requis pour les méthodes LIBRE et NORMALISÉE. Nous obtenons des résultats similaires pour la méthode SEBER, voir §3.6.2.

4.6 Conclusions et perspectives

Conclusions. Nous avons étudié le problème de l'optimisation non-linéaire de la matrice fondamentale. Les paramétrisations de cette dernière sont soit redondantes, par exemple la paramétrisation "épipôle + homographie" a 12 paramètres, soit minimales mais nécessitent plusieurs cartes.

Nous avons proposé la *factorisation totale* de la matrice fondamentale, reliée aux épipôles et à la transformation épipolaire 1D. Cette factorisation totale est obtenue en étendant la transformation épipolaire au 2D via la *transformation épipolaire étendue* que nous introduisons. Cette transformation affecte aux droites épipolaires de l'image de gauche, des points sur les droites épipolaires correspondantes dans l'image de droite. Ce formalisme permet de construire la plupart des paramétrisations minimales de la matrice fondamentale proposées



$\mathcal{A}$ $\mathcal{B}$

$\mathcal{C}$ $\mathcal{D}$

FIG. 4.8 – Images utilisées pour comparer les différentes méthodes. Différentes paires sont formées pour obtenir des épipôles proches des images ou à l'infini. Les 60 correspondances de points ont été données manuellement et sont montrées en blanc sur les images.

dans la littérature dont nous donnons un état de l'art complet.

Nous avons ensuite proposé la *représentation orthonormale* de la matrice fondamentale. C'est une sur-paramétrisation complète pouvant être mise à jour localement par une loi de mise à jour simple. A partir de cette représentation, on construit des algorithmes d'estimation non-linéaire de la matrice fondamentale basés sur le nombre minimal de 7 paramètres. Nous construisons explicitement la minimisation des distances point-droite épipolaire et de l'erreur de reprojection (ajustement de faisceaux). Tout autre critère peut de même être minimisé en utilisant la représentation orthonormale de la matrice fondamentale.

Nous avons comparé l'ajustement de faisceaux basé sur la représentation orthonormale à d'autres algorithmes d'ajustement de faisceaux et à d'autres paramétrisations de la matrice fondamentale. Nos conclusions sont que les méthodes basées sur un nombre minimal de paramètres ont des coûts moins élevés que les autres, et donnent des résultats de même précision.

La méthode s'étend naturellement au cas multi-vues en modélisant deux vues de référence par une matrice fondamentale, et les vues restantes par des matrices de projection, exprimées dans une base canonique de l'espace, induite par la matrice fondamentale.

Perspectives. Une des applications de la transformation épipolaire étendue est la suivante. Hartley [83] propose un algorithme linéaire permettant d'estimer la matrice fondamentale étant donné les coordonnées d'un épipôle, avec $m \geq 5$ points. Cette méthode estime implicitement l'autre épipôle et la transformation épipolaire. Elle est basée sur la contrainte épipolaire, équation (2.42), $\mathbf{q}'^{\mathsf{T}}\mathsf{F}\mathbf{q} = 0$, reécrite en $\mathbf{q}'^{\mathsf{T}}[\mathbf{e}']_\wedge \mathsf{H}_{\boldsymbol{\Pi}}\mathbf{q} = 0$, si l'on suppose que c'est $\boldsymbol{e}'$ qui est connu. En remplaçant $[\mathbf{e}']_\wedge$ par sa décomposition en valeurs singulières, Hartley [83] montre qu'un système linéaire est induit sur 5 variables représentant l'épipôle $\boldsymbol{e}$ et la transformation épipolaire. Nous renvoyons à l'annexe B, où cette méthode est appliquée à l'estimation d'une géométrie dynamique, pour plus de détails. Basé sur la transformation épipolaire étendue, on peut imaginer généraliser la méthode au cas où les deux épipôles sont connus. En remplaçant F par sa factorisation totale (4.5), la contrainte épipolaire devient $\mathbf{q}'^{\mathsf{T}}[\mathbf{e}']_\wedge \mathsf{G}_{\boldsymbol{\Pi},\boldsymbol{s}}[\mathbf{e}]_\wedge \mathbf{q} = 0$. En remplaçant $[\mathbf{e}]_\wedge$ et $[\mathbf{e}']_\wedge$ par leurs décompositions en valeurs singulières, on obtient un système linéaire pour 3 paramètres représentant la transformation épipolaire, qui possède une solution en général unique pour $m \geq 3$ correspondances de points.

Une perspective de travail sur l'estimation de la géométrie épipolaire est la minimisation quasi-linéaire des critères d'erreur utilisés ci-dessus – distances point-droite épipolaire et erreur de reprojection – par repondération itérative de systèmes linéaires. Quelques auteurs [86, 233] ont envisagé une telle approche, en négligeant la contrainte non-linéaire de rang de la matrice fondamentale. Les résultats expérimentaux montrent qu'en l'état, cette méthode n'améliore que très peu l'erreur donnée par l'algorithme des huit points et qu'elle est loin d'une minimisation non-linéaire standard telle l'algorithme de Levenberg-Marquardt. Zhang [233] conclut que c'est le fait d'ignorer la contrainte de rang qui en est la cause. Nous proposons d'incorporer cette contrainte au premier ordre à chaque itération d'un schéma quasi-linéaire. Nous avons implanté et validé une méthode similaire pour la triangulation multi-vues de droites représentées par des coordonnées de Plücker : ces coordonnées sont soumises à une contrainte bilinéaire, que nous linéarisons et incorporons au système linéaire résolu à chaque étape, grâce à une des méthodes d'estimation linéaire sous contraintes linéaires décrites en §2.6.4, chapitre 2. Dans le cas des droites, cette méthode donne des résultats équivalents à une minimisation de l'erreur par l'algorithme de Levenberg-Marquardt. Dans le cas de la matrice fondamentale, des résultats préliminaires permettent d'envisager des conclusions similaires.

CHAPITRE

5

RECONSTRUCTION DE PLANS, DE POINTS ET DE CAMÉRAS

Ce chapitre aborde le problème de l'incorporation de contraintes géométriques au processus de reconstruction. Plus particulièrement, nous considérons des contraintes de *multi-coplanarité*, c'est à dire l'appartenance d'un point à des plans. De telles contraintes sont omniprésentes en environnement humain.

Nous abordons les quatre aspects suivants du problème. Le premier est celui de la gestion des contraintes. Ces dernières peuvent entraîner l'interdépendance de tous les paramètres de la structure. Nous proposons de faire l'hypothèse qu'un point n'appartienne pas à plus de trois plans. La structure est alors modélisée par des *plans indépendants* et des *points contraints sur ces plans*.

Le deuxième problème étudié est celui du calcul d'une reconstruction initiale. La solution classique est de reconstruire les points en ignorant les contraintes, d'y ajuster des plans puis de corriger la position des points afin qu'ils satisfassent les contraintes. Nous proposons de reconstruire les plans puis les points. Dans le cas de deux vues, la reconstruction d'un plan est effectuée par estimation d'une homographie. Nous généralisons cette approche au cas multi-vues puis introduisons des méthodes de *triangulation contrainte par multi-coplanarité*. Ces méthodes sont linéaires et minimisent une erreur exprimée dans les images.

Le troisième problème étudié est celui de l'ajustement de faisceaux plan par morceaux. Ce problème est formulé comme la minimisation de l'erreur de reprojection sous contraintes non-linéaires. Les algorithmes existants sont sous-optimaux ou de complexité élevée. Nous proposons une approche basée sur une *paramétrisation minimale*. Ainsi formulé, le problème est résolu par un algorithme d'ajustement de faisceaux standard.

L'utilisation de contraintes géométriques stabilise la reconstruction. Cependant, il n'existe pas de surface réelle parfaitement plane. Notre but est de déterminer expérimentalement, jusqu'à quelle déviation de planarité, l'utilisation des contraintes de multi-coplanarité améliore la reconstruction.

Nous utilisons des données simulées et réelles pour comparer nos algorithmes à l'ajustement de faisceaux standard et à d'autres algorithmes de reconstruction utilisant la planarité par morceaux.

Ces travaux ont été effectués en collaboration avec Peter Sturm et partiellement avec Radu Horaud et ont été publiés dans [28, 29, 30, 31] pour le cas spécifique de deux vues et dans [21, 25] pour le cas multi-vues.

5.1 Introduction

Le problème étudié dans ce chapitre est l'incorporation au processus de reconstruction de contraintes géométriques sur la structure. L'utilisation de contraintes géométriques est étudiée depuis de nombreuses années en photogrammétrie, dans des contextes divers. Par exemple, Strunz [191] montre que des contraintes géométriques améliorent la précision du calcul d'orientation et Mikhail [144] en utilise pour réduire le nombre de points de contrôle nécessaires.

Notre but est d'améliorer la précision d'une reconstruction de points et de caméras à l'aide de contraintes de coplanarité entre les points reconstruits. De telles configurations sont très souvent observées en environnement humain, autant dans des scènes d'intérieur que d'extérieur, par exemple sur les scènes de type architectural. Les scènes comportant de nombreuses contraintes de coplanarité entre les points sont dites *planes par morceaux*. Cette expression sous-entend que la surface de la scène peut être facilement modélisée par un ensemble de plans. Un exemple typique de scène plane par morceaux est montré en figure 5.1. Notons qu'en environnement humain, des contraintes métriques, de type parallélisme ou orthonormalité, sont souvent applicables entre les plans modélisés.

(a) (b) (c)

FIG. 5.1 – (a) et (b) : paire d'images d'une scène plane par morceaux, appelée "game system I" et (c) : modèle métrique reconstruit, consistant de 109 points et de 7 plans. Le rendu texturé est obtenu par correction perspective des textures des images originales vers l'image synthétique.

La planarité par morceaux d'une scène donne des contraintes géométriques fortes, desquelles nous pouvons attendre une amélioration de la précision de la reconstruction par rapport à une reconstruction éparse de points. Une structure plane par morceaux est plus contrainte qu'une structure éparse, ce qui entraîne une réduction du nombre de degrés de liberté à estimer. Par exemple, les 109 points reconstruits à partir des images de la figure 5.1 donnent une structure à 327 degrés de liberté dans le cas non contraint, et à 188 degrés de liberté dans le cas d'une modélisation plane par morceaux, soit une réduction de plus de 42%.

Le problème de la modélisation et de la reconstruction de scènes de type architectural est étudié par de nombreux auteurs. Certains, voir par exemple [50, 95, 117, 190], proposent l'utilisation de primitives géométriques de haut niveau, telles le cube, le prisme ou la sphère, comme briques de base pour la modélisation interactive. Des logiciels commerciaux tels `canoma` ou `photomodeller` ont été élaborés à partir de ces travaux. L'utilisation de telles primitives, couplée à des contraintes géométriques de positionnement, semble naturel pour la modélisation de scènes de type architectural. Ces primitives permettent la modélisation d'une scène avec peu d'interactions. Les inconvénients sont la perte d'une forme de souplesse dans le type de scènes pouvant être modélisées et le fait que la détection de primitives de haut niveau est très difficilement automatisable.

En revanche, l'utilisation de plans permet une modélisation assez souple d'un grand nombre de scènes. Les aspects de modélisation automatique par *détection de plans* sont étudiés entre autres dans [9, 33, 55, 68, 113, 182, 183]. Nous proposons une méthode de détection de plans dans l'annexe A. Nous y présentons un état de l'art détaillé des différentes approches. Notons que Cornou et al. [44] et Wilczkowiak et al. [227] utilisent un ensemble de relations géométriques simples entre points (égalité, orthogonalité et coplanarité). Des méthodes

de calibrage en ligne des caméras spécifiques au cas des scènes planes par morceaux sont proposées dans [74, 132, 217, 231].

Problèmes abordés et contributions. Ce chapitre aborde le problème de la reconstruction de plans, de points et de caméras étant données des correspondances de points et des relations de multi-coplanarité (appartenance d'un point à un ou plusieurs plans).

Une des interrogations essentielles pour la résolution d'un tel problème concerne la fonction de coût. Les points induisent une erreur de reprojection *géométrique*, dépendant des points image mesurés et prédits par la reconstruction. Les plans, en tant que surfaces, induisent une erreur de type *photométrique*, mesurable par comparaison de l'intensité de pixels. L'erreur de type photométrique est très sensible aux conditions d'observation. Nous renvoyons à l'annexe A (où une erreur photométrique est utilisée à des fins de détection de plans) pour plus de détails sur ce point. La plupart des auteurs, voir §5.3.2, utilisent des formulations proches de l'erreur de reprojection induite par les points. Nous suivons cette approche et utilisons l'erreur de reprojection. Les plans sont reconstruits grâce aux contraintes géométriques. La reconstruction est optimale, au sens du maximum de vraisemblance, sous les conditions définies au chapitre 3, si l'erreur de reprojection est minimisée et si les contraintes de multi-coplanarité sont satisfaites.

Nous abordons les quatre points suivants : (i) la gestion des contraintes géométriques, (ii) l'initialisation de la reconstruction, (iii) l'ajustement de faisceaux plan par morceaux et (iv) l'évaluation de conditions d'observation et de planarité pour lesquelles la prise en compte des contraintes de multi-coplanarité améliore la précision de la reconstruction, par rapport à une reconstruction non contrainte de points.

La gestion des contraintes géométriques est un des points clefs lors de la conception d'une méthode de reconstruction plane par morceaux. Certains auteurs proposent de n'utiliser que des contraintes de coplanarité simple (chaque point appartient à un plan au plus). C'est en effet la contrainte la plus utilisée en pratique. Nous proposons de faire l'hypothèse qu'un point n'est impliqué au plus que dans trois contraintes de coplanarité. Cette hypothèse permet de modéliser la structure comme un ensemble de plans et un ensemble de points sur ces plans, indépendants les uns des autres, ce qui simplifie grandement la gestion des contraintes, sans toutefois réduire notoirement la généralité de la méthode, et permet la mise en œuvre de solutions efficaces pour la reconstruction. Nous montrons expérimentalement que prendre en compte les contraintes de multi-coplanarité avec deux et trois plans fait une différence significative par rapport à l'hypothèse de coplanarité simple.

La solution classique pour la reconstruction initiale est de reconstruire les points par une méthode non contrainte standard, d'ajuster des plans, puis de corriger la position des points afin qu'ils satisfassent les contraintes géométriques. Cette dernière étape introduit un biais. Nous proposons de l'éviter en reconstruisant d'abord les plans puis les points directement sur les plans. Pour ce faire, nous étendons la reconstruction de plans à partir de deux vues par estimation d'une homographie au cas multi-vues. La reconstruction de points sur ces plans est ensuite effectuée par des méthodes de triangulation contrainte par multi-coplanarité que nous proposons. Toutes les méthodes proposées sont linéaires. Elles minimisent des erreurs exprimées dans les images, dont la forme est proche de celle de l'erreur de reprojection, et traitent toutes les données uniformément.

L'ajustement de faisceaux plan par morceaux est un problème de minimisation non-linéaire sous contraintes non-linéaires. La plupart des solutions existantes sont sous-optimales. Certaines ne minimisent pas l'erreur de reprojection, d'autres ne considèrent les contraintes de multi-coplanarité que partiellement ou approximativement. Une pratique courante est l'ajout de termes de pénalité à l'erreur de reprojection, permettant de faire tendre la structure estimée vers une configuration plane par morceaux. Cette solution change la fonction de coût. La structure ainsi reconstruite n'est qu'approximativement plane par morceaux. La difficulté principale est que les contraintes sont non-linéaires. Nous proposons une approche basée sur une paramétrisation minimale, c'est à dire dont le nombre de paramètres est égal au nombre de degrés de liberté physiques de la structure. Avec cette paramétrisation, le problème peut être résolu par un algorithme d'ajustement de faisceaux non contraint standard. Notre paramétrisation permet la résolution éparse des équations normales.

Nos contributions sont d'abord formulées pour le cas spécifique de deux vues, où une base de reconstruction canonique permet d'obtenir des paramétrisations et des algorithmes simples, puis sont généralisées au cas multi-vues.

Finalement, nous menons une étude expérimentale, dont le but est de déterminer dans quelles configurations

d'observation et pour quels types de scènes, l'incorporation de contraintes de multi-coplanarité améliore la précision de la reconstruction, par rapport à une reconstruction éparse de points, même si la surface réelle n'est qu'approximativement plane. Cette étude révèle que dans beaucoup de cas pratiques, l'incorporation des contraintes améliore la précision.

Organisation du chapitre. Nous donnons une modélisation algébrique du problème en §5.2. Nous y étudions la gestion des contraintes de multi-coplanarité, les degrés de liberté et la reconstructabilité d'une structure plane par morceaux. En §5.3, nous donnons un état de l'art et situons l'approche proposée. Nous développons nos contributions pour l'initialisation et l'ajustement de faisceaux pour le cas de deux vues et le cas multi-vues en §§5.4 et 5.5 respectivement. Le chapitre se termine par des résultats expérimentaux sur des données simulées et réelles, où nous donnons le schéma d'un système complet de reconstruction, suivis d'une ouverture vers d'autres types de contraintes géométriques puis de nos conclusions et perspectives.

5.2 Modélisation du problème

Le problème abordé est celui de la reconstruction conjointe de points et de plans. Comme annoncé dans l'introduction, l'erreur photométrique induite par les plans n'est pas prise en compte. Ces derniers sont déterminés par les contraintes de multi-coplanarité. Les observations portent uniquement sur les points reconstruits. La fonction de coût conduisant à une solution au maximum de vraisemblance est l'erreur de reprojection définie par l'équation (3.1) et rappelée ci-dessous :

$$\mathcal{Q}(\dots, \hat{\mathbf{q}}_{ij}, \dots) \;=\; \sum_{i,j} d^2_{PP}(\mathbf{q}_{ij}, \hat{\mathbf{q}}_{ij}),$$

en supposant sans perte de généralité que tous les points sont visibles dans toutes les vues. Les $\boldsymbol{q}_{ij}$ sont les points image mesurés et les $\hat{\boldsymbol{q}}_{ij}$ sont les points correspondants prédits par la reconstruction. La structure est modélisée par un ensemble de plans $\boldsymbol{\Pi}_1, \dots, \boldsymbol{\Pi}_z$ et de points $\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_m$. Afin de modéliser les contraintes de multi-coplanarité, nous définissons les constantes c_{jl} par :

$$c_{jl} \;=\; \begin{cases} 1 & \text{si } \boldsymbol{Q}_j \in \boldsymbol{\Pi}_l \\ 0 & \text{sinon.} \end{cases}$$

La contrainte $\boldsymbol{Q}_j \in \boldsymbol{\Pi}_l$ s'exprime algébriquement par :

$$\mathbf{Q}_j^\top \mathbf{\Pi}_l \;=\; 0.$$

Notons qu'en espace métrique, avec la normalisation $\mathbf{Q}_j^\top = (\bar{\mathbf{Q}}_j^\top \;\; 1)$ et $\|\bar{\mathbf{\Pi}}_l\| = 1$, $\mathbf{Q}_j^\top \mathbf{\Pi}_l$ donne la distance Euclidienne (au signe près) entre le point $\boldsymbol{Q}_j$ et le plan $\boldsymbol{\Pi}_l$, voir l'équation (2.14). Le problème de la reconstruction plane par morceaux consiste à minimiser la fonction de coût $\mathcal{Q}$ sur les paramètres des caméras, des points et des plans, sous les contraintes de multi-coplanarité $c_{jl}\mathbf{Q}_j^\top \mathbf{\Pi}_l = 0$.

5.2.1 Gestion des contraintes de multi-coplanarité

Les contraintes de multi-coplanarité sont l'appartenance d'un point à un certain nombre de plans. Ceci définit un réseau de contraintes. Si l'on se place dans le cas général, n'importe quels points ou plans peuvent être dépendants. En effet, lorsque trois points appartiennent à un plan, la position de ce dernier est alors déterminée. Il exerce donc une contrainte sur les autres points qu'il contient, et ainsi de suite. Imaginons l'expression de ces contraintes sous forme d'un graphe. Ce dernier peut comporter des cycles et des redondances, ce qui peut, selon les cas, excessivement compliquer la gestion des contraintes.

Pour ces raisons, nous ne modélisons que les contraintes d'appartenance d'un point à un, deux ou trois plans, c'est à dire $\forall j, (\sum_l c_{jl}) \leq 3$. Dans ce cas, la modélisation suivante du problème est possible : chaque plan est modélisé indépendamment, alors que les points sont soumis aux contraintes de multi-coplanarité. Une

telle hypothèse est réaliste car il est rare d'observer des points sur plus de trois plans (sur l'exemple "game system I", les points sont sur un ou deux plans, voir figure 5.2). Elle permet de simplifier grandement la gestion des contraintes et par suite leur modélisation algébrique. Les méthodes proposées dans ce chapitre peuvent toutefois être adaptées au cas général. Ceci est abordé en §5.6, où l'on donne plus de détails sur la modélisation algébrique des contraintes de multi-coplanarité avec plus de trois plans, ainsi que des liens avec certaines contraintes métriques entre les plans modélisés. Cette hypothèse permet la reconstruction indépendante de chaque plan puis de points sur ces plans, voir §5.3.1. Elle permet de même de mettre en œuvre une solution efficace pour la résolution des équations normales lors de la phase d'ajustement de faisceaux, voir §5.4.2.2.

5.2.2 Etude du nombre de degrés de liberté

Nous donnons une expression du nombre de degrés de liberté d'une structure non contrainte et d'une structure plane par morceaux. Ces expressions permettent d'apprécier la réduction du nombre de degrés de liberté apportée par l'utilisation des contraintes. Un point de l'espace a 3 degrés de liberté. Une structure de m points a donc $\mathcal{D} = 3 \times m$ degrés de liberté. Les points modélisés peuvent appartenir à 0, 1, 2 ou 3 plans. De tels points ont un nombre de degrés de liberté variable, dépendant du nombre de plans les contenants : chaque contrainte de coplanarité "enlève" un degré de liberté au point. On obtient donc la formule suivante pour le nombre de degrés de liberté $\mathcal{E}(s)$ d'un point appartenant à s plans en position générale :

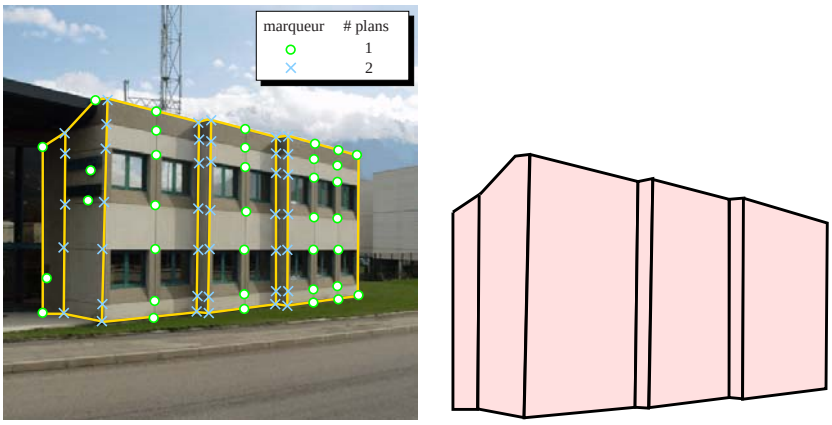
$$\mathcal{E}(s) \quad = \quad 3 - s.$$

Un point non contraint a 3 degrés de liberté, un point sur 1 plan en a 2 et ainsi de suite. Cette formule donne des degrés de liberté négatifs aux points sur plus de 3 plans, ce qui reflète le fait qu'un ou plusieurs des plans correspondants sont alors contraints. Notons $m(s)$ le nombre de points sur s plans et z le nombre de plans modélisés. Le nombre de degrés de liberté total de la structure est donné par :

$$\mathcal{E} \quad = \quad 3z + \sum_{s=0}^{3} \mathcal{E}(s)m(s) \quad = \quad 3z + \sum_{s=0}^{3} (3-s)m(s).$$

Nous ne traitons pas les cas dégénérés : nous supposons que les plans sont tous différents et qu'aucun triplet de plans ne forme de faisceau.

Comparons les nombres de degrés de liberté $\mathcal{D}$ d'une structure non contrainte et $\mathcal{E}$ d'une structure plane par morceaux sur l'exemple basé sur les images montrées en figure 5.1. La figure 5.2 fait apparaître une modélisation avec 109 points et 7 plans. Le nombre de degrés de liberté d'une modélisation non contrainte est donc $\mathcal{D} = 3 \times 109 = 327$. Parmi ces points, 69 sont sur 1 plan et 40 sont sur 2 plans. Le nombre de degrés de liberté



(a) (b)

FIG. 5.2 – Modélisation de la scène "game system I" : (a) montre un sous-ensemble représentatif des 109 points utilisés et avec (b) les 7 plans modélisés.

d'une modélisation plane par morceaux est donc $\mathcal{E} = 3 \times 7 + 2 \times 69 + 1 \times 40 = 188$, soit une réduction de plus

# vues	# plans	non contrainte	contrainte	af
0	0	non	non	non
	1		non	
	2		non	
	≥ 3		oui	
1	0	non	non	non
	1		oui	non
	≥ 2		oui	oui
≥ 2	quelconque	oui	oui	oui

TAB. 5.1 – Résumé de la reconstructabilité des différents types de points. “# vues” et “# plans” donnent respectivement le nombre de vues dans lesquelles le point est visible et le nombre de plans auxquels il appartient. “non contrainte” indique la reconstructabilité lorsque les plans ne sont pas encore reconstruits et “contrainte” indique la reconstructabilité lorsque les plans sont disponibles. “af” indique que le point apporte une redondance d’information utile à l’ajustement de faisceaux plan par morceaux.

de 42%. Notons que cet exemple ne contient pas de points non contraints ou de points sur trois plans. D’autres exemples impliquant de tels points, ainsi que des détails sur la reconstruction de la scène “game system I” sont donnés en §5.7.2.

5.2.3 Reconstructabilité

Nous examinons la reconstructabilité d’une structure plane par morceaux en supposant que celle des caméras est possible. Dans le cas non contraint, les points peuvent être reconstruits si ils sont visibles dans deux vues au moins. Ils apportent ensuite une redondance d’information utile pour la phase d’ajustement de faisceaux. Dans le cas d’une structure plane par morceaux, le cas des plans, puis différents cas pour les points doivent être considérés. Ces cas sont résumés dans le tableau 5.1. Un plan est reconstructible si au moins trois points qu’il contient peuvent eux-mêmes être reconstruits sans contrainte géométrique, c’est à dire si ils sont visibles dans au moins deux vues. La modélisation plane par morceaux permet de reconstruire les points non observés si et seulement si ils appartiennent à trois plans. Ces points n’apportent pas de redondance d’information utile pour la phase d’ajustement de faisceaux car trois plans en position générale ont un unique point d’intersection. Les points visibles dans une seule vue peuvent être reconstruits si ils sont sur un plan ou plus. Lorsqu’ils sont sur deux plans ou plus, ils entrent alors dans l’ajustement de faisceaux, car ils amènent une redondance d’information utile.

5.3 Etat de l’art et approche proposée

Nous donnons un état de l’art puis introduisons l’approche proposée et la situons par rapport aux autres approches. Nous abordons les phases d’initialisation puis d’ajustement de faisceaux plan par morceaux.

5.3.1 Initialisation

5.3.1.1 Etat de l’art

Une des premières questions posée pour l’initialisation concerne l’ordre de reconstruction des différentes entités, parmi les caméras, les plans et les points, c’est à dire les principales étapes du processus. La difficulté est de produire des méthodes linéaires. La plupart des auteurs utilisent l’ordre montré sur la figure 5.3 (a) : caméras, points en ignorant les contraintes géométriques, ajustement de plans aux groupes de points censés être coplanaires, et enfin correction de la position des points pour qu’ils satisfassent les contraintes de multi-coplanarité. Cette approche est simple mais a plusieurs défauts, voir ci-dessous.

D'autres ordres de reconstruction sont possibles. On peut difficilement parler de la reconstruction des points avant que celle des caméras soit estimée. Par contre, on peut calculer les homographies induites par les différents groupes de points coplanaires, puis estimer la matrice fondamentale (c'est à dire une reconstruction des caméras) à partir de ces homographies, comme proposé par Luong et Faugeras [129] et Sinclair et al. [184]. Szeliski et Torr [200] montrent que ces algorithmes sont très instables. Notons que Zelnik-Manor et Irani [232] exhibent un jeu de contraintes sur les homographies 2D induites par un ensemble de plans. Elles montrent que ces contraintes peuvent être utilisées, entre autres, pour améliorer des homographies 2D estimées indépendamment. Certains auteurs, par exemple Rother [169] ou Sun et al. [199], proposent d'intégrer entre autres des contraintes de coplanarité à des méthodes de factorisation de points et de caméras[1]. Rother [169] factorise directement les plans, les points et les caméras, alors que l'idée de Sun et al. [199] est de projetter la matrice de mesure sur les contraintes.

Nous passons en revue différentes techniques pour les phases de reconstruction caméras - points non contraints - ajustement de plans et correction des points.

Reconstruction des caméras. La reconstruction des caméras est souvent effectuée de manière conjointe à celle des points. Nous renvoyons au chapitre 3, §3.2 pour un exposé plus détaillé des différentes méthodes. La reconstruction des caméras est identique quel que soit l'ordre de reconstruction utilisé car elle en est la première étape.

Triangulation des points. La triangulation non contrainte des points est une phase sensible. Il serait préférable de pouvoir profiter des contraintes géométriques pour la stabiliser, même s'il existe des algorithmes de triangulation optimaux, voir [92].

Ajustement des plans. Une fois les caméras et points reconstruits, et connaissant les relations de multi-coplanarité, on peut former des ensembles de points censés être coplanaires. Pour chaque ensemble, on peut ajuster un plan en fonction d'un critère d'erreur. Si $\boldsymbol{Q}$ est un point sur le plan $\boldsymbol{\Pi}$, alors la contrainte $\mathbf{Q}^\top \mathbf{\Pi} = 0$ doit être satisfaite.

La méthode classique est d'empiler ces équations pour tous les points censés être sur le plan $\boldsymbol{\Pi}$. On obtient alors un système linéaire. Le critère d'erreur minimisé par sa résolution au sens des moindres carrés est la somme des carrés des distances algébriques entre les points et le plan. Notons que si la reconstruction est métrique, et en assurant une normalisation appropriée des équations, alors ce critère est équivalent à la somme des carrés des distances Euclidiennes entre les points et le plan. Cette solution peut donner de bons résultats en espace métrique mais le critère est dépourvu de sens en espace projectif. Dans ce cas, il faudrait minimiser un critère basé image. Un tel critère est facile à formuler dans le cas de deux vues, en faisant intervenir l'homographie de plan induite. Nous proposons une méthode de ce type en §5.4.1.2 pour le cas de deux vues et l'étendons au cas multi-vues en §5.5.1.2.

Notons que l'étape d'ajustement de plans est importante car elle conditionne l'étape suivante, la correction des points.

Correction des points. La correction des points consiste à trouver, pour chaque point reconstruit, un point le plus proche possible et satisfaisant les contraintes de multi-coplanarité. Une telle correction introduit inévitablement un biais. En espace métrique, on peut imaginer projeter orthogonalement les points sur les plans les contenant. En espace projectif, un telle projection n'existe pas.

Une solution possible pour la correction d'un point sur un plan est la suivante. Notons qu'elle ne prend pas toutes les données en compte de manière uniforme, mais qu'elle minimise un critère de correction basé image. Pour un point $\boldsymbol{Q}$ censé être sur un plan $\boldsymbol{\Pi}$, on considère sa projection dans une des images. Le point 3D contraint est obtenu par intersection du rayon de projection émanant de cette image avec le plan $\boldsymbol{\Pi}$. Afin d'obtenir le meilleur résultat possible, on mesure l'erreur de reprojection induite par la correction du point

[1] Les méthodes de factorisation sont décrites en §3.2.1.2, chapitre 3.

dans toutes les images où ce point est visible. On essaye itérativement toutes les images afin de sélectionner la meilleure solution.

Pour le cas d'un point sur deux plans, la solution suivante minimise l'erreur de reprojection dans une des images. Soit $\boldsymbol{Q}$ un point censé être sur deux plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi'}$. Pour assurer que le point corrigé se trouve bien sur les deux plans, on projette orthogonalement le point image sur la projection de la droite représentant l'intersection des deux plans. On reconstruit ensuite le point 3D comme dans le cas précédent, par intersection du rayon de vue issue du point image corrigé et avec un des deux plans. Cette solution n'est pas satisfaisante dans la mesure où toutes les observations des points ne sont pas prises en compte de manière uniforme.

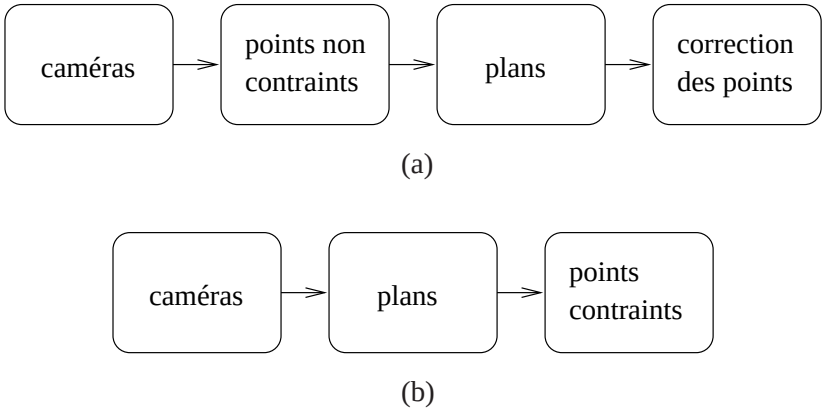


FIG. 5.3 – Illustration des différentes étapes de l'initialisation, (a) : une approche classique et (b) : approche proposée.

5.3.1.2 Approche proposée

La reconstruction initiale est effectuée de la manière suivante, illustrée sur la figure 5.3 (b) : les caméras sont reconstruites, puis les plans et enfin les points, sous contraintes de multi-coplanarité. Cet ordre induit des fonctions de coût dont la forme est proche de celle de l'erreur de reprojection, et limite donc le biais introduit par rapport à la solution optimale. Cet ordre de reconstruction est possible grâce au fait que les points n'appartiennent pas à plus de trois plans. Les plans sont indépendants les uns des autres et les points dépendent des plans. Notons que la détermination des caméras est identique au cas précédent. Nous renvoyons donc à la section 5.3.1.1 pour leur reconstruction.

Reconstruction des plans. La détermination des plans avant la reconstruction des points fait appel à un critère d'erreur basé image, par exemple dans le cas de deux vues, cela correspond à l'estimation de l'homographie de plan induite. La généralisation de cette idée au cas multi-vues induit une fonction de coût de forme plus complexe. Nous proposons une méthode de ce type en §5.4.1.2 pour le cas de deux vues et l'étendons au cas multi-vues en §5.5.1.2. Le critère que nous formulons est linéaire, basé image et ne dépend pas des points 3D, mais seulement des points image. C'est cette propriété qui nous permet de reconstruire les plans sans reconstruire les points au préalable. L'utilisation d'un critère basé image a donc deux avantages principaux : il peut être utilisé indifféremment en espaces métrique et projectif, et il permet la reconstruction des plans avant les points. Ces derniers peuvent alors être reconstruits directement sur les plans.

Triangulation de points sur plans. La reconstruction de points contraints d'appartenir à des plans est effectuée par des méthodes de *triangulation contrainte par multi-coplanarité* que nous proposons. Ces méthodes sont basées sur des paramétrisations des points incluant les contraintes de multi-coplanarité. Elles sont linéaires et minimisent une erreur basée image. Elles sont présentées en §5.4.1 pour le cas de deux vues et en §5.5.1 pour le cas multi-vues.

5.3.2 Ajustement de faisceaux plan par morceaux

L'ajustement de faisceaux consiste à minimiser l'erreur de reprojection sur la structure et les caméras. Le chapitre 3 examine différentes techniques d'optimisation non-linéaire utilisées pour résoudre ce problème. Chaque itération requiert la résolution d'équations normales. Le coût d'une telle optimisation est réduit en utilisant des techniques de résolution éparse des équations normales (voir chapitre 3, §3.6.3). L'état de l'art donné ci-dessous montre que les méthodes existantes n'induisent pas toutes des équations normales avec une structure éparse par bloques, contrairement à l'approche proposée.

5.3.2.1 Etat de l'art

L'ajustement de faisceaux plan par morceaux se formule comme la minimisation de l'erreur de reprojection sous contraintes de multi-coplanarité : c'est un problème de minimisation non-linéaire sous contraintes non-linéaires. Les méthodes existantes sont sous-optimales ou font appel à des techniques d'optimisation sous contraintes compliquées. Le tableau 5.2 résume les caractéristiques de différentes méthodes.

méthode	optimale	paramètres	# paramètres	éparse	réf.
Szeliski et Torr	non	$\mathbf{Q}_1, \ldots, \mathbf{Q}_m, \mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_z$	$3m + 3z$	non	[200]
McGlone	non	idem	idem	non	[138]
Xu et al.	oui	$\hat{\mathbf{q}}_1, \ldots, \hat{\mathbf{q}}_m, \mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_z$	$2m + 3z$	oui	[231]
Bondyfalat et Bougnoux	oui	voir légende	minimal	non	[35]
McLauchlan et al.	oui	$\mathbf{Q}_1, \ldots, \mathbf{Q}_m, \mathbf{\Pi}_1, \ldots, \mathbf{\Pi}_z, \gamma_{jl}$	$3m + 3z +$ # cont	oui	[139]

TAB. 5.2 – Résumé de l'état de l'art des méthodes d'ajustement de faisceaux permettant de prendre en compte une structure plane par morceaux. m est le nombre de points et z le nombre de plans. La colonne "éparse" indique si une résolution éparse par bloques des équations normales est possible. Toutes ces méthodes, à l'exception de la méthode de Xu et al., fonctionnent avec plusieurs vues et modélisent la multi-coplanarité. Les paramètres de la structure obtenus par Bondyfalat et Bougnoux sont des fonctions polynomiales non triviales des paramètres "naturels" (coordonnées homogènes) des points et des plans. La méthode de McLauchlan et al. produit un nombre de paramètres faisant intervenir le nombre de contraintes modélisées noté "# cont". Son approche est semblable aux techniques standards utilisées en photogrammétrie.

Une pratique courante pour la reconstruction plane par morceaux, au moins lorsque deux vues sont considérées, est de faire intervenir les homographies de plan induites entre les deux images par les plans modélisés. Ceci permet de paramétrer les points sur un plan, le cas de multi-coplanarité le plus courant, par un seul point image. Xu et al. [231] proposent un système complet de reconstruction et calibrage en ligne de deux caméras observant une scène plane par morceaux. La phase finale est un ajustement de faisceaux. Seule la coplanarité simple est modélisée : chaque point $\boldsymbol{Q}_j$ est paramétré par un point de la première image $\hat{\boldsymbol{q}}_{1j}$. La fonction de coût minimisée est :

$$\sum_{j,l} c_{jl} \left(d^2(\mathbf{q}_{1j}, \hat{\mathbf{q}}_{1j}) + d^2(\mathbf{q}_{2j}, \mathsf{H}_{\boldsymbol{\Pi}_l} \hat{\mathbf{q}}_{1j}) \right) ,$$

où $\mathsf{H}_{\boldsymbol{\Pi}_l}$ est l'homographie de plan induite par le plan $\boldsymbol{\Pi}_l$. Cette méthode n'est optimale que dans le cas où seuls des points sur un plan sont traités. Son extension au cas multi-vues est difficile. Les travaux présentés ci-dessous montrent qu'il est plus approprié d'utiliser une paramétrisation basée sur des entités 3D pour le cas multi-vues. Une méthode de reconstruction inspirée de celle de Xu et al. est comparée aux nôtres en §5.7.

Szeliski et Torr [200] incorporent des contraintes de multi-coplanarité dans un système d'ajustement de faisceaux métrique par l'ajout de pénalités fortement pondérées à l'erreur de reprojection $\mathcal{Q}$. Leur méthode est inspirée de la méthode pour la résolution de systèmes linéaires sous contraintes linéaires présentée en §2.6.4.1, chapitre 2. La fonction de coût minimisée est :

$$\mathcal{Q} + \gamma \sum_{j,l} c_{jl} \mathbf{Q}_j^\mathsf{T} \mathbf{\Pi}_l .$$

Le facteur γ est fixé à une valeur très élevée, de l'ordre de $2^{60} \approx 10^{20}$. La structure obtenue n'est qu'approximativement plane par morceaux. De plus, des termes d'un ordre de grandeur important entrent dans la fonction de coût, ce qui modifie l'estimation et déséquilibre l'estimateur non-linéaire d'un point de vue numérique. Cette méthode est comparée aux nôtres en §5.7.

McGlone [138] propose l'incorporation de contraintes géométrique à un système d'ajustement de faisceaux assez complet. Les contraintes sont traitées comme des mesures. La fonction de coût minimisée par le système est similaire à celle de Szeliski et Torr. La différence est que chaque contrainte est pondérée indépendamment par l'introduction de facteurs γ_{jl} :

$$\mathcal{Q} + \sum_{j,l} \gamma_{jl} c_{jl} \mathbf{Q}_j^{\mathsf{T}} \mathbf{\Pi}_l. \tag{5.1}$$

Les facteurs γ_{jl} sont déterminés en fonction de l'incertitude associée à chaque contrainte. La manière dont sont gérées les contraintes, qui peuvent être très générales (colinéarité, parallélisme de droites ou de plans, etc.), entraîne la violation de la structure éparse par bloques des équations normales. L'optimisation non-linéaire devient alors très coûteuse.

Bondyfalat et Bougnoux [35] modélisent des contraintes plus générales que la multi-coplanarité (incidence, parallélisme, orthogonalité, etc. entre points, droites et plans). Une technique d'élimination polynomiale des variables redondantes, la méthode de Ritt-Wu, permet d'obtenir une paramétrisation minimale. Le processus est très coûteux et viole la structure des équations normales. De plus, la résolution des équations polynomiales conduit à des ambiguïtés car plusieurs solutions sont possibles (la fonction de paramétrisation n'est pas bijective). La paramétrisation minimale est donc hautement non-linéaire, ce qui réduit les performances de l'optimisation non-linéaire. Les résultats sont optimaux, sous réserve de la bonne convergence de l'estimateur.

McLauchlan et al. [139] utilisent des multiplicateurs de Lagrange pour modéliser les contraintes. La fonction de coût utilisée est celle de McGlone [138], équation (5.1). La différence est que les poids γ_{jk} sont les multiplicateurs de Lagrange et sont donc estimés au cours de l'optimisation non-linéaire. McLauchlan et al. montrent comment préserver la structure éparse des équations normales. Cette méthode conduit à un résultat optimal, dans le sens où les contraintes sont respectées et l'erreur de reprojection minimisée. Les inconvénients de la méthode sont qu'en pratique l'utilisation de multiplicateurs de Lagrange soulève certains problèmes, notamment celui de l'initialisation. De plus, chaque contrainte modélisée ajoute une inconnue au système, augmentant le coût de sa résolution en conséquence. Cette approche est similaire aux techniques standards d'ajustement de faisceaux en photogrammétrie, où des multiplicateurs de Lagrange sont systématiquement utilisés pour modéliser des contraintes. C'est l'approche photogrammétrique. Un ouvrage de référence est le recueil d'Atkinson [6], et plus particulièrement pour ce problème, la contribution de Cooper et Robson [43]. Van Den Heuvel et Vosselman [221] utilisent l'approche photogrammétrique pour la reconstruction de bâtiments à l'aide de contraintes géométriques variées, coplanarité, colinéarité, etc. Shan et al. [176] proposent une approche de ce style, basée sur la linéarisation des contraintes, et un calcul analytique permettant de déterminer les multiplicateurs de Lagrange correspondant. Ces derniers ne sont donc plus inclus dans l'optimisation. Cette approche est attirante, mais présente certaines faiblesses, notamment l'approximation linéaire des contraintes et le fait que la détermination analytique des multiplicateurs de Lagrange ajoute des termes non-linéaires de forme non triviale à la fonction de coût. Wilczkowiak et al. [227] éliminent les variables redondantes avec un jeu de règles d'élimination.

Debevec et al. [50] reconstruisent des primitives géométrique de haut niveau (cube, pyramide, etc). Leur fonction de coût s'exprime en termes des segments prédits par le modèle et des segments mesurés. Elle conduit à un résultat sous-optimal.

Si l'on veut résumer cet état de l'art en se concentrant sur le critère de l'optimalité statistique des méthodes, seules les approches de Bondyfalat et Bougnoux [35] et McLauchlan et al. [139] peuvent théoriquement conduire à un résultat optimal, mais accroîssent notoirement le coût de calcul par rapport à une reconstruction non contrainte. Ces méthodes font intervenir des techniques d'optimisation non-linéaire compliquées. Le nombre de degrés de liberté d'une structure plane par morceaux est généralement inférieur à celui d'une structure non contrainte. Il devrait donc être possible de réduire le coût de calcul de la reconstruction non contrainte par la modélisation d'une structure plane par morceaux, et non de l'augmenter.

5.3.2.2 Approche proposée

Notre approche est de transformer le problème original en un problème d'ajustement de faisceaux non contraint, qui peut alors être résolu par une méthode standard. Ceci est effectué par une paramétrisation minimale de la structure incorporant les contraintes de multi-coplanarité. Cette approche a l'avantage que le nombre d'inconnues est minimal, par opposition par exemple aux méthodes basées sur des multiplicateurs de Lagrange, où chaque contrainte ajoute une inconnue au système. Notre paramétrisation préserve la structure éparse par bloques des équations normales.

5.4 Le cas de deux vues

Nous traitons le cas particulier de deux vues. Des solutions spécifiques très simples peuvent être mises en œuvre pour la paramétrisation et la reconstruction initiale. Nous définissons une *paramétrisation minimale, générale et linéaire* de la structure et des caméras. Cette paramétrisation est utilisée dans les phases d'initialisation et d'ajustement de faisceaux. Elle garantit que les contraintes géométriques sont satisfaites. Par exemple, un point sur un plan est exprimé avec seulement deux paramètres. Les coordonnées du point 3D correspondant sont données en fonction des ces deux paramètres et du plan support. Cette section concernant uniquement le cas de deux vues, nous n'y utilisons pas l'indice i pour désigner la vue, mais plutôt une notation "nue" pour la première vue et avec $'$ pour la deuxième (par exemple, la correspondance de point $\boldsymbol{q}_{1j} \leftrightarrow \boldsymbol{q}_{2j}$ est écrite $\boldsymbol{q}_j \leftrightarrow \boldsymbol{q}'_j$).

5.4.1 Initialisation

Nous suivons l'ordre de reconstruction caméras - plans - points sur plans. Nous proposons des méthodes pour la reconstruction des plans puis des méthodes de *triangulation contrainte par multi-coplanarité* pour une reconstruction directe de points satisfaisant les contraintes.

5.4.1.1 Caméras

La reconstruction de deux caméras non calibrées est basée sur l'estimation de la matrice fondamentale F. Nous renvoyons au chapitre 4 dédié à ce problème. Nous formons la *réalisation canonique non singulière* de la matrice fondamentale, donnée par les équations (2.43) et (2.47), et rappelée ci-dessous :

$$\begin{aligned} \mathsf{P} &\sim (\ \mathrm{I} \quad \mathbf{0}) \\ \mathcal{P}' &\sim (\ \mathcal{H} \quad \mathbf{e}'), \end{aligned}$$

où $\boldsymbol{e'}$ est le deuxième épipôle et $\mathcal{H} = ([\mathbf{e}']_\wedge \mathsf{F} + \mathbf{e}'\mathbf{e}^\mathsf{T})/\|\mathbf{e}'\|^2$ est l'homographie canonique non singulière donnée par l'équation (2.46). Nous notons P' la deuxième matrice de projection ($\mathsf{P}' = \mathcal{P}'$). C'est la forme particulière de ces matrices de projection qui permet une paramétrisation simple de la structure.

5.4.1.2 Plans

Nous disposons de la reconstruction des caméras et pour chaque plan $\boldsymbol{\Pi}$, de correspondances de points $\boldsymbol{q}_j \leftrightarrow \boldsymbol{q}'_j$. Le plan $\boldsymbol{\Pi}$ induit une homographie $\mathsf{H}_{\boldsymbol{\Pi}}$ telle que :

$$\mathbf{q}'_j \sim \mathsf{H}_{\boldsymbol{\Pi}} \mathbf{q}_j.$$

Dans la base canonique, tout plan $\boldsymbol{\Pi}$ peut être représenté par son équation réduite $\bar{\mathbf{\Pi}}$, voir §2.11. L'homographie $\mathsf{H}_{\boldsymbol{\Pi}}$ a la forme donnée par l'équation (2.51) et rappelée ci-dessous :

$$\mathsf{H}_{\boldsymbol{\Pi}} \sim \mathcal{H} - \mathbf{e}'\bar{\mathbf{\Pi}}^\mathsf{T}.$$

Une solution classique pour la reconstruction du plan $\boldsymbol{\Pi}$ est d'estimer les 9 coefficients de $\mathsf{H}_{\boldsymbol{\Pi}}$, puis d'en extraire $\bar{\mathbf{\Pi}}$ via l'équation (2.51). La présence de bruit dans les données entraîne que les points ne se correspondent

pas parfaitement. L'homographie ne peut donc être estimée que de manière approchée et la redondance d'information est exploitée par la minimisation d'une fonction de coût. Cette fonction de coût est choisie pour que l'estimation soit linéaire, typiquement, voir par exemple [93, §3] et la section 2.6.1 du chapitre 2 :

$$\sum_j d^2_{A_PP}(\mathbf{q}'_j, \mathsf{H}_{\boldsymbol{\Pi}}\mathbf{q}_j), \tag{5.2}$$

où d_{A_PP} est une distance algébrique dans le plan Euclidien donnée par l'équation (2.18). Cette solution pour la reconstruction de $\boldsymbol{\Pi}$ a plusieurs défauts :

- ▷ un minimum de 4 correspondances de points sont nécessaires, alors que 3 devraient être suffisants ;
- ▷ la fonction de coût (5.2) n'est pas symétrique par rapport aux rôles joués par les deux images : l'erreur est minimisée dans la deuxième image seulement ;
- ▷ le résultat n'est pas direct : l'homographie 2D estimée ne correspond pas exactement à une homographie de plan, d'où la nécessité d'en extraire $\bar{\boldsymbol{\Pi}}$.

Pour remédier à ces lacunes, nous proposons de minimiser une version symétrique de la fonction de coût (5.2), sous la contrainte que $\mathsf{H}_{\boldsymbol{\Pi}}$ soit une homographie de plan :

$$\min_{\mathsf{H}_{\boldsymbol{\Pi}}} \mathcal{C}_{\boldsymbol{\Pi}}(\mathsf{H}_{\boldsymbol{\Pi}}) \text{ sous contrainte que } \mathsf{H}_{\boldsymbol{\Pi}} \text{ soit une homographie de plan},$$

avec :

$$\mathcal{C}_{\boldsymbol{\Pi}}(\mathsf{H}_{\boldsymbol{\Pi}}) \;=\; \sum_j \left(d^2_{A_PP}(\mathbf{q}'_j, \mathsf{H}_{\boldsymbol{\Pi}}\mathbf{q}_j) + d^2_{A_PP}(\mathbf{q}_j, \mathsf{H}^{-1}_{\boldsymbol{\Pi}}\mathbf{q}'_j)\right). \tag{5.3}$$

La solution que nous proposons permet de résoudre ce problème de manière linéaire. Une solution exacte est donnée par 3 correspondances de points et une solution minimisant (5.3) est obtenue pour plus, ce qui est consistant avec le fait qu'un plan est défini par 3 points. La difficulté principale est d'exprimer $\mathsf{H}_{\boldsymbol{\Pi}}$ et $\mathsf{H}^{-1}_{\boldsymbol{\Pi}}$ de manière linéaire en fonction d'un même jeu de paramètres. Pour ce faire, nous exploitons le fait que $\mathsf{H}_{\boldsymbol{\Pi}}$ soit une homographie de plan. L'équation (2.51) permet d'exprimer $\mathsf{H}_{\boldsymbol{\Pi}}$ de manière linéaire en terme de l'équation réduite $\bar{\boldsymbol{\Pi}}$ du plan. Nous proposons l'équation suivante, donnant $\mathsf{H}^{-1}_{\boldsymbol{\Pi}}$, comme une fonction linéaire de $\bar{\boldsymbol{\Pi}}$:

$$\mathsf{H}^{-1}_{\boldsymbol{\Pi}} \;\sim\; \mathbf{e}\bar{\boldsymbol{\Pi}}^\mathsf{T}\mathcal{H}^{-1} + (1 - \bar{\boldsymbol{\Pi}}^\mathsf{T}\mathbf{e})\mathcal{H}^{-1}. \tag{5.4}$$

Cette équation est équivalente à la formule non-linéaire de Sherman-Morrison à l'échelle près, voir par exemple [93, §12.5.2]. On peut facilement la vérifier en effectuant le produit de $\mathsf{H}_{\boldsymbol{\Pi}}$ et $\mathsf{H}^{-1}_{\boldsymbol{\Pi}}$ par les équations (2.51) et (5.4). Notons que la condition $\mathbf{e}' = \mathcal{H}\mathbf{e}$ est nécessaire, et pas seulement $\mathbf{e}' \sim \mathcal{H}\mathbf{e}$. Cette condition est vérifiée par l'homographie canonique non singulière, équation (2.46), définissant la base de reconstruction. Notons aussi que l'inverse de l'homographie de référence $\mathcal{H}^{-1}$ est nécessaire, d'où la nécessité d'une base canonique non singulière. Remplaçons $\mathsf{H}_{\boldsymbol{\Pi}}$ et $\mathsf{H}^{-1}_{\boldsymbol{\Pi}}$ par leurs expressions données par les équations (2.51) et (5.4) respectivement dans la fonction de coût (5.3) :

$$\mathcal{C}_{\boldsymbol{\Pi}}(\mathsf{H}_{\boldsymbol{\Pi}}) \;=\; \sum_j \left(d^2_{A_PP}(\mathbf{q}'_j, (\mathcal{H} - \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T})\mathbf{q}_j) + d^2_{A_PP}(\mathbf{q}_j, (\mathbf{e}\bar{\boldsymbol{\Pi}}^\mathsf{T}\mathcal{H}^{-1} + (1 - \bar{\boldsymbol{\Pi}}^\mathsf{T}\mathbf{e})\mathcal{H}^{-1})\mathbf{q}'_j)\right).$$

Après développement, et en utilisant la définition (2.18) de la distance algébrique d_{A_PP} dans le plan Euclidien, nous obtenons :

$$\mathcal{C}_{\boldsymbol{\Pi}}(\mathsf{H}_{\boldsymbol{\Pi}}) \;=\; \sum_j \|\mathsf{S}[\mathbf{q}'_j]_\wedge(\mathcal{H}\mathbf{q}_j - \mathbf{e}'\bar{\boldsymbol{\Pi}}^\mathsf{T}\mathbf{q}_j)\|^2 + \|\mathsf{S}[\mathbf{q}_j]_\wedge(\mathbf{e}\bar{\boldsymbol{\Pi}}^\mathsf{T}\mathcal{H}^{-1}\mathbf{q}'_j + (1 - \bar{\boldsymbol{\Pi}}^\mathsf{T}\mathbf{e})\mathcal{H}^{-1}\mathbf{q}'_j)\|^2,$$

qui se reécrit en :

$$\mathcal{C}_{\boldsymbol{\Pi}}(\mathsf{H}_{\boldsymbol{\Pi}}) \;=\; \sum_j \|\mathsf{S}[\mathbf{q}'_j]_\wedge\mathbf{e}'\mathbf{q}_j^\mathsf{T}\bar{\boldsymbol{\Pi}} - \mathsf{S}[\mathbf{q}'_j]_\wedge\mathcal{H}\mathbf{q}_j\|^2 + \|\mathsf{S}[\mathbf{q}_j]_\wedge(\mathcal{H}^{-1}\mathbf{q}'_j\mathbf{e}^\mathsf{T} - \mathbf{e}\mathbf{q}_j'^\mathsf{T}\mathcal{H}^{-\mathsf{T}})\bar{\boldsymbol{\Pi}} - \mathsf{S}[\mathbf{q}_j]_\wedge\mathcal{H}^{-1}\mathbf{q}'_j\|^2,$$

et sous forme matricielle :

$$\mathcal{C}_{\boldsymbol{\Pi}} \quad = \quad \sum_j \|\mathsf{A}_j \bar{\boldsymbol{\Pi}} - \mathbf{b}_j\|^2,$$

avec :

$$\mathsf{A}_{j\,(4\times3)} \quad = \quad \begin{pmatrix} \mathsf{S}[\mathbf{q}'_j]_\wedge \mathbf{e}' \mathbf{q}_j^\mathsf{T} \\ \mathsf{S}[\mathbf{q}_j]_\wedge (\mathcal{H}^{-1}\mathbf{q}'_j \mathbf{e}^\mathsf{T} - \mathbf{e}\mathbf{q}'^\mathsf{T}_j \mathcal{H}^{-\mathsf{T}}) \end{pmatrix} \quad \text{et} \quad \mathbf{b}_{j\,(4\times1)} \quad = \quad \begin{pmatrix} \mathsf{S}[\mathbf{q}'_j]_\wedge \mathcal{H}\mathbf{q}_j \\ \mathsf{S}[\mathbf{q}_j]_\wedge \mathcal{H}^{-1}\mathbf{q}'_j \end{pmatrix}.$$

Chacune des m correspondances fournit donc 4 équations dont 1 seulement est indépendante. L'équation réduite de plan minimisant $\mathcal{C}_{\boldsymbol{\Pi}}$ est donnée par la résolution au sens des moindres carrés du système suivant :

$$\mathsf{A}_{(4m\times3)} \bar{\boldsymbol{\Pi}}_{(3\times1)} \quad = \quad \mathbf{b}_{(4m\times1)} \quad \text{avec} \quad \mathsf{A}^\mathsf{T} \quad = \quad (\ldots \ \mathsf{A}_j^\mathsf{T} \ \ldots) \quad \text{et} \quad \mathbf{b}^\mathsf{T} \quad = (\ldots \ \mathbf{b}_j^\mathsf{T} \ \ldots), \qquad (5.5)$$

par exemple en utilisant une pseudo-inverse de A : $\bar{\boldsymbol{\Pi}} = \mathsf{A}^\dagger \mathbf{b}$, voir la section 2.6.3 du chapitre 2. Notons qu'il est possible d'appliquer un schéma de résolution quasi-linéaire, comme décrit en §2.6.2, chapitre 2. La fonction de coût minimisée par un tel algorithme se déduit de l'équation (5.3), car il suffit de remplacer la distance algébrique d_{A_PP} par la distance Euclidienne correspondance d_{PP} :

$$\sum_j \left(d^2_{PP}(\mathbf{q}'_j, \mathsf{H}_{\boldsymbol{\Pi}} \mathbf{q}_j) + d^2_{PP}(\mathbf{q}_j, \mathsf{H}^{-1}_{\boldsymbol{\Pi}} \mathbf{q}'_j) \right).$$

5.4.1.3 Points non contraints – Triangulation

Soit $\boldsymbol{Q}$ un point 3D non soumis à des contraintes géométriques. La reconstruction d'un tel point ne dépend pas de la modélisation plane par morceaux de la structure. La méthode de choix est la triangulation de Hartley et Sturm [85]. Cette méthode minimise l'erreur de reprojection. Elle implique le calcul des racines d'un polynôme de degré 6. Cette méthode donne des résultats optimaux, mais elle est difficile à étendre aux cas de reconstruction contrainte examinés ci-après. Nous présentons une méthode de triangulation linéaire qui s'étend ensuite aux cas plans par morceaux. On paramétrise le point comme proposé par Hartley et Zisserman [93], voir la section 2.10.1.3 du chapitre 2 :

$$\mathbf{Q} \quad \sim \quad \begin{pmatrix} \hat{\mathbf{q}} \\ \\ \lambda \end{pmatrix} \quad \sim \quad \begin{pmatrix} \hat{q}_1 \\ \hat{q}_2 \\ 1 \\ \lambda \end{pmatrix}.$$

On considère la fonction de coût suivante :

$$\mathcal{C}_{\boldsymbol{Q}}(\mathbf{Q}) \quad = \quad d^2_{A_PP}(\mathbf{q}, \mathsf{P}\mathbf{Q}) + d^2_{A_PP}(\mathbf{q}', \mathsf{P}'\mathbf{Q}), \qquad (5.6)$$

qui peut être minimisée en résolvant le système suivant au sens des moindres carrés :

$$\mathsf{A}_{(4\times4)} \mathbf{Q}_{(4\times1)} \quad = \quad \mathbf{0}_{(4\times1)} \quad \text{avec} \quad \mathsf{A} \quad \sim \quad \begin{pmatrix} \mathsf{S}[\mathbf{q}]_\wedge \\ \mathsf{S}[\mathbf{q}']_\wedge \mathsf{P}' \end{pmatrix}. \qquad (5.7)$$

Notons qu'il est possible d'utiliser le schéma de résolution quasi-linéaire décrit en §2.6.2, chapitre 2. La fonction de coût minimisée est donnée en remplaçant la distance algébrique d_{A_PP} par la distance Euclidienne d_{PP} correspondante dans l'équation (5.6) :

$$d^2_{PP}(\mathbf{q}, \mathsf{P}\mathbf{Q}) + d^2_{PP}(\mathbf{q}', \mathsf{P}'\mathbf{Q}).$$

Cette équation correspond à l'erreur de reprojection pour les points dans deux vues, voir la section 3.3.1 du chapitre 3. La méthode de triangulation quasi-linéaire est donc équivalente à la méthode de triangulation de Hartley et Sturm [85], sous réserve de sa convergence au minimum global de la fonction de coût.

5.4.1.4 Points sur un plan – Triangulation contrainte par coplanarité

Soit $\boldsymbol{Q}$ un point contraint sur un plan $\boldsymbol{\Pi}$. Si $\boldsymbol{\Pi}$est fixé, $\boldsymbol{Q}$ n'a plus que 2 degrés de liberté car l'équation de coplanarité $\mathbf{Q}^\mathsf{T}\mathbf{\Pi} = 0$ doit être satisfaite. Nous proposons une paramétrisation minimale de $\boldsymbol{Q}$ (avec 2 variables) puis une méthode de *triangulation contrainte par coplanarité*, c'est à dire permettant de reconstruire le point $\boldsymbol{Q}$ directement sur le plan $\boldsymbol{\Pi}$. La paramétrisation et la reconstruction sont illustrés sur la figure 5.4.

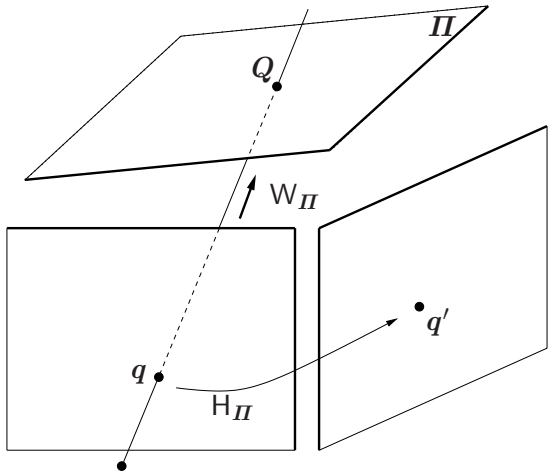


FIG. 5.4 – Paramétrisation minimale par le point image $\boldsymbol{q}$ et reconstruction par la matrice $\mathsf{W}_{\boldsymbol{\Pi}}$ d'un point $\boldsymbol{Q}$ sur un plan $\boldsymbol{\Pi}$. L'opération $\mathbf{Q} \sim \mathsf{W}_{\boldsymbol{\Pi}}\mathbf{q}$ modélise l'intersection du rayon de vue associé à $\boldsymbol{q}$ avec le plan $\boldsymbol{\Pi}$.

Paramétrisation. Une paramétrisation minimale naturelle d'un tel point est sa reprojection dans une des images. Notre choix se porte sur sa reprojection dans la première image, $\hat{\mathbf{q}}$. Sa reprojection dans la deuxième image est donnée par application de l'homographie de plan induite par $\boldsymbol{\Pi}$:

$$\hat{\mathbf{q}}' \sim \mathsf{H}_{\boldsymbol{\Pi}}\hat{\mathbf{q}}.$$

Le point 3D $\boldsymbol{Q}$ correspondant est obtenu en intersectant le rayon de projection émanant de la première vue avec le plan $\boldsymbol{\Pi}$. Comme dans le cas précédent, nous avons $\mathbf{Q}^\mathsf{T} \sim (\hat{q}_1 \;\; \hat{q}_2 \;\; 1 \;\; \lambda)$. Seul λ reste à déterminer. Utilisons l'équation de coplanarité :

$$\mathbf{Q}^\mathsf{T}\mathbf{\Pi} = \hat{\mathbf{q}}^\mathsf{T}\bar{\mathbf{\Pi}} + \lambda = 0,$$

d'où :

$$\lambda = -\hat{\mathbf{q}}^\mathsf{T}\bar{\mathbf{\Pi}},$$

ce qui donne la paramétrisation minimale linéaire suivante :

$$\mathbf{Q}_{(4\times1)} \sim \mathsf{W}_{\boldsymbol{\Pi}\,(4\times3)}\hat{\mathbf{q}}_{(3\times1)} \quad \text{avec} \quad \mathsf{W}_{\boldsymbol{\Pi}} = \begin{pmatrix} \mathrm{I}_{(3\times3)} \\ -\bar{\mathbf{\Pi}}^\mathsf{T} \end{pmatrix}. \tag{5.8}$$

Cette équation se vérifie facilement en calculant les reprojections de $\boldsymbol{Q}$ et en vérifiant qu'elles correspondent bien à $\hat{\boldsymbol{q}}$ et $\hat{\boldsymbol{q}}'$.

Reconstruction. Sturm [195] puis Chum et Padjla [42] montrent que la solution optimale du problème de la reconstruction d'un point sur un plan (minimisant l'erreur de reprojection) nécessite la résolution d'un polynôme de degré 8. Nous proposons la solution suivante. Considérons le système linéaire (5.7) du cas non contraint et remplaçons $\mathbf{Q}$ par la paramétrisation (5.8) :

$$\mathsf{A}\mathsf{W}_{\boldsymbol{\Pi}}\hat{\mathbf{q}} = 0. \tag{5.9}$$

Un tel système peut être résolu au sens des moindres carrés pour retrouver $\hat{\mathbf{q}}$ puis en déduire $\mathbf{Q} \sim \mathsf{W}_{\boldsymbol{\Pi}} \hat{\mathbf{q}}$. La fonction de coût minimisée est la même que pour le cas non contraint, elle est donnée par l'équation (5.6). On peut utiliser le schéma de résolution quasi-linéaire comme précédemment. L'algorithme est alors optimal, car il minimise l'erreur de reprojection, tout en garantissant la contrainte de coplanarité.

5.4.1.5 Points sur deux plans – Triangulation contrainte par double coplanarité

Soit $\boldsymbol{Q}$ un point contraint sur deux plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi'}$. Ce point doit se trouver sur $\boldsymbol{L}$, la droite d'intersection des deux plans, dont la projection $\boldsymbol{l}$ dans la première image a des coordonnées données par l'équation (2.52) :

$$\mathbf{l} \;=\; \bar{\mathbf{\Pi}} - \bar{\mathbf{\Pi}}'.$$

Ces notations sont illustrées sur la figure 5.5. Si l'on base la paramétrisation de $\boldsymbol{Q}$ sur $\hat{\boldsymbol{q}}$, sa reprojection dans

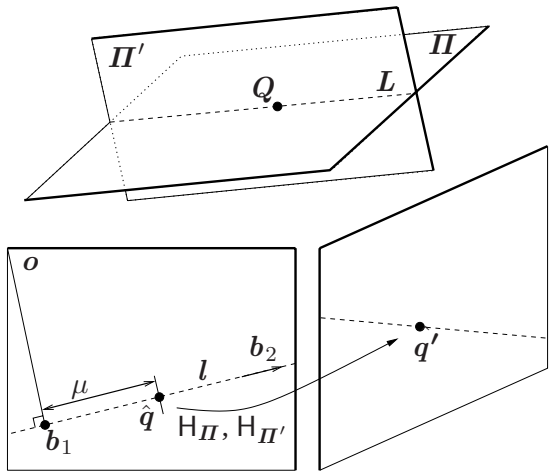


FIG. 5.5 – Paramétrisation d'un point $\boldsymbol{Q}$ appartenant à deux plans, par un point image $\hat{\boldsymbol{q}}$, contraint sur l'image $\boldsymbol{l}$ de la droite d'intersection des deux plans. Le paramètre μ est une paramétrisation minimale de $\boldsymbol{Q}$. Les coordonnées de $\boldsymbol{Q}$ sont données en intersectant le rayon de vue issu de $\hat{\boldsymbol{q}}$ avec un des deux plans.

la première image, alors $\hat{\boldsymbol{q}} \in \boldsymbol{l}$ est équivalent à $\boldsymbol{Q} \in \boldsymbol{\Pi}$ et $\boldsymbol{Q} \in \boldsymbol{\Pi}'$.

Paramétrisation. Le problème est de paramétrer le point image $\hat{\boldsymbol{q}}$, sur une droite image $\boldsymbol{l}$, avec un seul paramètre. L'idée est de choisir deux points $\boldsymbol{b}_1$ et $\boldsymbol{b}_2$ sur $\boldsymbol{l}$ et de s'en servir comme base pour exprimer tous les points sur $\boldsymbol{l}$. On veut que ce choix soit systématique, c'est à dire qu'il ne dépende pas de la position de $\boldsymbol{l}$. Par exemple, un choix naturel est l'intersection de $\boldsymbol{l}$ avec les bords de l'image, disons les bords verticaux. Ce choix ne peut pas être systématique : si $\boldsymbol{l}$ est verticale, ses deux intersections avec l'axe des y seront à l'infini. Nous proposons le choix systématique suivant. Pour $\boldsymbol{b}_1$, la projection orthogonale de $\boldsymbol{o}$, l'origine du repère image, sur $\boldsymbol{l}$, et pour $\boldsymbol{b}_2$, le point à l'infini de $\boldsymbol{l}$, c'est à dire sa direction. Ces points sont définis par :

$$\mathbf{b}_1 \;\sim\; \begin{pmatrix} \bar{\mathbf{o}} - \frac{\mathbf{o}^\top \mathbf{l}}{\|\mathbf{l}\|^2} \bar{\mathbf{l}} \\ 1 \end{pmatrix} \;\sim\; \begin{pmatrix} -l_1 l_3 \\ -l_2 l_3 \\ l_1^2 + l_2^2 \end{pmatrix}$$

$$\mathbf{b}_2 \;\sim\; \mathbf{l} \wedge \mathbf{l}_\infty \;\sim\; \begin{pmatrix} l_1 \\ l_2 \\ l_3 \end{pmatrix} \wedge \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \;\sim\; \begin{pmatrix} l_2 \\ -l_1 \\ 0 \end{pmatrix}.$$

Si $\mathbf{l}$ est normalisé tel que[2] $l_1^2 + l_2^2 = 1$, $\mathbf{b}_1$ s'écrit :

$$\mathbf{b}_1 \sim \begin{pmatrix} -l_1 l_3 \\ -l_2 l_3 \\ 1 \end{pmatrix}.$$

Le point $\hat{\boldsymbol{q}}$ n'est pas à l'infini et peut donc être paramétré par un scalaire μ tel que :

$$\hat{\mathbf{q}} \sim \mathbf{b}_1 + \mu \mathbf{b}_2.$$

Le point 3D $\boldsymbol{Q}$ correspondant est donné comme dans le cas précédent en intersectant le rayon de projection issu de $\hat{\boldsymbol{q}}$ avec un des plans $\boldsymbol{\Pi}$et $\boldsymbol{\Pi'}$. En utilisant pour ce faire l'équation (5.8), on obtient la paramétrisation minimale linéaire suivante :

$$\mathbf{Q} \sim \mathsf{W}_{\boldsymbol{\Pi}} \hat{\mathbf{q}} \sim \mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_1 + \mu \mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_2. \tag{5.10}$$

Reconstruction. Considérons le système linéaire (5.7) du cas non contraint et remplaçons $\mathbf{Q}$ par la paramétrisation (5.10) :

$$\begin{aligned} \mathsf{A}(\mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_1 + \mu \mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_2) &= \mathbf{0} \\ \mathsf{A}\mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_2 \mu &= -\mathsf{A}\mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_1. \end{aligned}$$

Une solution analytique pour la résolution de ce système au sens des moindres carrés est donnée en faisant le produit scalaire avec le vecteur $\mathsf{A}\mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_2$, ce qui conduit à :

$$\mu = -\frac{\mathbf{b}_2^\mathsf{T} \mathsf{W}_{\boldsymbol{\Pi}}^\mathsf{T} \mathsf{A}^\mathsf{T} \mathsf{A} \mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_1}{\mathbf{b}_2^\mathsf{T} \mathsf{W}_{\boldsymbol{\Pi}}^\mathsf{T} \mathsf{A}^\mathsf{T} \mathsf{A} \mathsf{W}_{\boldsymbol{\Pi}} \mathbf{b}_2}. \tag{5.11}$$

La fonction de coût minimisée est la même que pour le cas non contraint, elle est donnée par l'équation (5.6). On peut utiliser le schéma de résolution quasi-linéaire comme précédemment. L'algorithme est alors optimal, car il minimise l'erreur de reprojection, tout en garantissant les contraintes de coplanarité.

5.4.1.6 Points sur trois plans – Intersection de trois plans

Un point $\boldsymbol{Q}$ contraint sur trois plans $\boldsymbol{\Pi}$, $\boldsymbol{\Pi}'$ et $\boldsymbol{\Pi}''$ n'a aucun degré de liberté et n'est donc pas paramétré. Ce cas est illustré sur la figure 5.6. L'équation (2.53) donne une expression analytique de la projection de $\boldsymbol{Q}$ dans la première image :

$$\hat{\mathbf{q}} \sim (\bar{\mathbf{\Pi}} \wedge \bar{\mathbf{\Pi}}') + (\bar{\mathbf{\Pi}}' \wedge \bar{\mathbf{\Pi}}'') + (\bar{\mathbf{\Pi}}'' \wedge \bar{\mathbf{\Pi}}).$$

Sa projection dans l'autre image est donnée par l'homographie de plan associée à l'un des trois plans. Ses coordonnées 3D sont données en intersectant le rayon de projection issu de$\hat{\boldsymbol{q}}$ avec un des trois plans, équation (5.8). Quel que soit le plan utilisé, on retombe sur l'équation (2.53) donnant les coordonnées de l'intersection de trois plans :

$$\mathbf{Q} \sim \mathsf{W}_{\boldsymbol{\Pi}} \hat{\mathbf{q}} \sim \begin{pmatrix} (\bar{\mathbf{\Pi}} \wedge \bar{\mathbf{\Pi}}') + (\bar{\mathbf{\Pi}}' \wedge \bar{\mathbf{\Pi}}'') + (\bar{\mathbf{\Pi}}'' \wedge \bar{\mathbf{\Pi}}) \\ (\bar{\mathbf{\Pi}}, \bar{\mathbf{\Pi}}', \bar{\mathbf{\Pi}}'') \end{pmatrix}, \tag{5.12}$$

où $(\mathbf{a}, \mathbf{b}, \mathbf{c})$ désigne le produit mixte des vecteurs $\mathbf{a}$, $\mathbf{b}$ et $\mathbf{c}$ défini par $(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \mathbf{a}^\mathsf{T}(\mathbf{b} \wedge \mathbf{c})$.

[2] Notons que comme $\boldsymbol{l}$ est visible dans l'image, elle est différente de la droite à l'infini et le terme $l_1^2 + l_2^2$ n'est jamais nul.

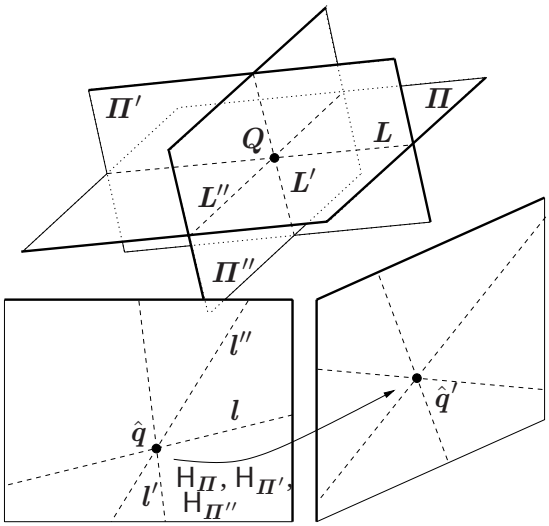


FIG. 5.6 – Un point appartenant à trois plans n'a pas de paramètres car il est déterminé (de manière unique si les trois plans sont en position générale) par l'intersection des trois plans.

5.4.1.7 Autres cas – Points observés dans moins de deux vues

Les autres cas sont les points visibles dans moins de deux vues. L'étude de reconstructabilité résumée dans le tableau 5.1 montre que certains de ces points peuvent être reconstruits sous certaines conditions. Ces cas se divisent en deux catégories, selon que les coordonnées du points sont déterminées de manière exacte (nombre de contraintes égal au nombre d'inconnues) ou en minimisant une fonction de coût (nombre de contraintes supérieur au nombre d'inconnues).

Nombre de contraintes égal au nombre d'inconnues. Les deux cas de figure sont les points non observés et sur trois plans et les points observés dans une vue et sur un plan. Ces points n'apportent pas de redondance d'information. Ils n'entrent pas dans l'ajustement de faisceaux et seront reconstruit à l'issue de cette dernière étape. Les points non observés sur trois plans sont définis par l'intersection de ces trois plans, donnée par l'équation (5.12). Les points observés dans une vue et sur un plan sont reconstruits en intersectant le rayon de vue induit par le point image observé et le plan, ce qu'effectue l'équation (5.8).

Nombre de contraintes supérieur au nombre d'inconnues. Les deux cas de figures sont les points observés dans une vue et sur deux ou trois plans. Ces points ne peuvent être reconstruit qu'après les plans. Ils amènent une redondance d'information utile à l'ajustement de faisceaux et sont donc initialisés avant cette étape. Les points sur deux plans sont reconstruits en minimisant l'erreur de reprojection, sous la contrainte de coplanarité, comme suit. Le point image est "corrigé" par projection orthogonale sur la droite image de l'intersection des deux plans. Le point 3D est ensuite obtenue en intersectant le rayon de vue issue du point image corrigé avec un des deux plans, par l'équation (5.8). Les points sur trois plans sont définis par l'intersection de ces trois plans, donnée par l'équation (5.12).

5.4.2 Ajustement de faisceaux plan par morceaux

Nous abordons le problème de l'ajustement de faisceaux plan par morceaux à partir de deux vues, défini comme la minimisation de l'erreur de reprojection, sous contraintes de multi-coplanarité. Comme annoncé en §5.3.2.2, nous transformons ce problème en un problème standard non contraint d'ajustement de faisceaux, grâce à l'incorporation des contraintes dans une *paramétrisation minimale* de la structure. Nous montrons que la paramétrisation minimale utilisée pour la reconstruction initiale peut être utilisée pour l'ajustement de faisceaux. Nous étudions ensuite la structure des équations normales et montrons comment une résolution

éparse par bloques est possible. Notons que le chapitre 3 présente quelques algorithmes standard d'ajustement de faisceaux.

5.4.2.1 Paramétrisation

Nous utilisons la paramétrisation proposée en §5.4.1 pour la reconstruction initiale. Cette paramétrisation est systématique et linéaire. Elle est basée sur des quantités image. Son utilisation dans un ajustement de faisceaux ne pose pas de problème. Elle est similaire pour les cas projectif et métrique. Entre ces deux cas, seule la paramétrisation du mouvement change. Pour le cas projectif, nous renvoyons au chapitre 4. Nous utilisons la représentation orthonormale de la matrice fondamentale. Pour le cas métrique, nous utilisons une matrice de rotation et un vecteur-3 de translation, voir par exemple [60] pour plus de détails. Le vecteur de paramètres est constitué des paramètres du mouvement (de la matrice fondamentale ou de la rotation/translation) et des paramètres de la structure, dans l'ordre plan, puis points sur 0, 1 et 2 plans.

5.4.2.2 Structure et résolution de équations normales

Chaque itération de l'ajustement de faisceaux nécessite de résoudre des équations normales, dont la taille est donnée par le nombre de paramètres. Des problèmes de grande taille peuvent être traités si les équations normales possèdent une structure éparse par bloques et si cette structure est prise en compte explicitement dans la résolution. Dans le cas contraire, chaque itération est très coûteuse. Les équations normales sont formées à partir de la matrice Jacobienne J de la fonction de coût et de l'approximation de Gauss-Newton de la matrice Hessienne $\mathsf{N} = \mathsf{J}^\mathsf{T}\mathsf{J}$. Pour plus de détails, on se reportera au chapitre 3, §3.6. Pour comprendre la résolution éparse par bloques des équations normales, il faut mettre en évidence la structure de N et donc celle de J. La

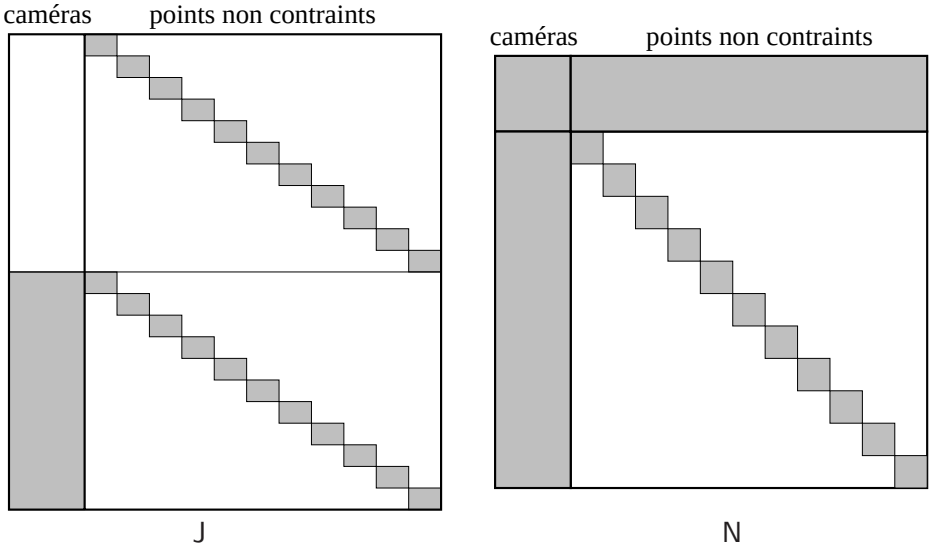


FIG. 5.7 – Structure des matrices Jacobienne J et Hessienne $\mathsf{N} = \mathsf{J}^\mathsf{T}\mathsf{J}$ (approximation de Gauss-Newton) pour l'ajustement de faisceaux standard. L'exemple montre une configuration avec 2 caméras représentées par la matrice fondamentale et un nuage de 11 points. La matrice Hessienne donne la structure des équations normales résolues à chaque étape de l'ajustement de faisceaux.

figure 5.7 montre la structure des matrices Jacobienne et Hessienne pour l'ajustement de faisceaux standard, avec deux caméras et un nuage de points. La résolution éparse des équations normales est basée sur la structure diagonale par bloques de la partie de la matrice Hessienne concernant les paramètres des points. Cette structure est due au fait que les résidus induits par chaque reprojection ne dépendent que des caméras et du point 3D correspondant. Chaque point a 3 paramètres et nécessite la résolution d'un système (3×3). Nous renvoyons au chapitre 3, §3.6.3 pour une présentation plus détaillée de la résolution éparse des équations normales.

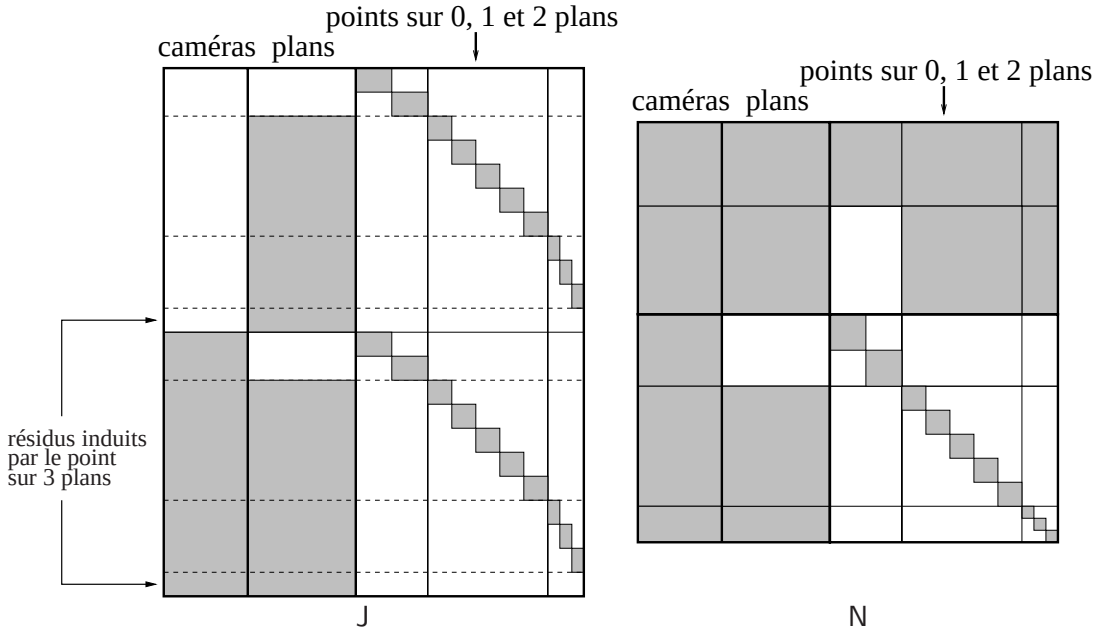


FIG. 5.8 – Structure des matrices Jacobienne J et Hessienne $\mathsf{N} = \mathsf{J}^T\mathsf{J}$ (approximation de Gauss-Newton) pour l'ajustement de faisceaux plan par morceaux proposé. L'exemple montre une configuration avec 2 caméras représentées par la matrice fondamentale, 3 plans et un nuage de 11 points, parmi lesquels 2 ne sont pas contraints, 5 sont sur 1 plan, 3 sont sur 2 plans et 1 est sur 3 plans. La matrice Hessienne donne la structure des équations normales résolues à chaque étape de l'ajustement de faisceaux. Notons que le point sur 3 plan n'a pas de paramètres, mais induit une erreur résiduelle, visible sur la matrice Jacobienne uniquement.

Revenons au cas de l'ajustement de faisceaux plan par morceaux. La figure 5.8 montre la structure des matrices Hessienne et Jacobienne, lorsque les paramètres sont rangés dans l'ordre caméra, plan et points sur 0, 1 et 2 plans. On observe que notre paramétrisation confère une structure éparse par bloques aux équations normales. Cette structure est légèrement différente de celle obtenue lors de l'ajustement de faisceaux non contraint. Examinons la structure de la matrice Jacobienne. La partie liée aux caméras ne change pas. Les plans n'influencent pas les résidus induit par les points non contraints, mais ceux induits par les points sur un plan ou plus. La partie liée aux points est similaire au cas non contraint, à la différence près que le nombre de paramètres pour chaque point dépend du nombre de plans le contenant.

Les équations normales du cas non contraint et plan par morceaux présentent donc quelques différences. Cependant, la résolution éparse est obtenue de la même manière. On identifie le bloque concernant les points du cas non contraint et du cas plan par morceaux, comme indiqué sur la figure 5.9. La résolution standard peut alors être appliquée. Chaque point sur s plans nécessite la résolution d'un système de taille $((3-s)\times(3-s))$.

Notons que c'est l'hypothèse qu'un point n'appartienne pas à plus de trois plans qui permet cette résolution éparse. Si un point appartenait à plus de trois plans, il entraînerait des dépendances entre points, ce qui violerait la forme diagonale par bloques dans la figure 5.8.

5.5 Le cas multi-vues

Le cas multi-vues ne peut pas être résolu directement par une paramétrisation minimale basée image, car on ne peut profiter des avantages de la base canonique. Il est possible de trouver une base de reconstruction projective où la première caméra est donnée par $\mathsf{P} \sim (\ \mathrm{I}\quad \mathbf{0})$, mais dû aux occultations, les points ne sont pas forcément visibles dans la vue correspondante. L'équation réduite des plans ne peut plus être utilisée : un plan peut contenir le centre optique de la première caméra tout en étant reconstructible car visible dans d'autres images. Une extension de la solution proposée pour deux vues peut cependant être mise en œuvre en exprimant

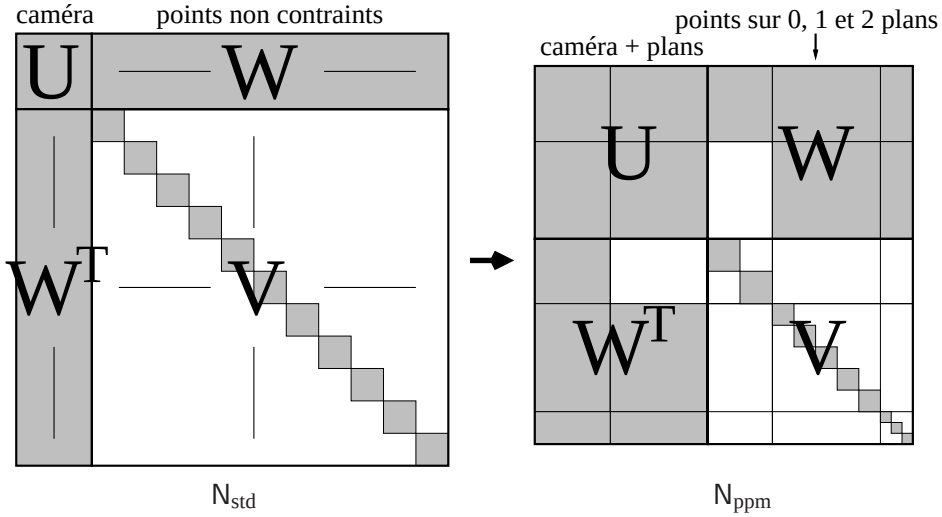


FIG. 5.9 – Structure des matrices Hessiennes (approximation de Gauss-Newton) N_{std} pour l'ajustement de faisceaux standard et N_{ppm} pour l'ajustement de faisceaux plan par morceaux proposé. En identifiant les 4 bloques comme indiqué ci-dessus, la résolution éparse standard des équations normales peut être appliquée au cas plan par morceaux.

chaque entité dans une base adéquate. Par exemple, si un plan est observé dans la vue i, son équation réduite peut être utilisée dans un repère centré sur la vue i. Trouver le changement de repère peut poser des problèmes pratiques en espace projectif.

Nous adoptons une stratégie différente, basée sur des choix de paramétrisation ne nécessitant pas la sélection d'une vue ou d'une base de reconstruction particulière. Nous étendons les méthodes d'initialisation proposées pour le cas de deux vues. Dans le cas de deux vues, la paramétrisation de la structure est identique pour les phases d'initialisation et d'ajustement de faisceaux. Dans le cas multi-vues, deux paramétrisations différentes sont utilisées.

5.5.1 Initialisation

On propose une extension du cas de deux vues. L'originalité de la méthode est que l'on préserve l'ordre d'initialisation caméras - plans - points contraints, produisant un résultat plus précis que la solution classique caméras - points non contraints - plans - points contraints.

5.5.1.1 Caméras

La reconstruction des caméras et éventuellement des points sans la prise en compte des contraintes de multi-coplanarité peut être effectuée par n'importe quel algorithme standard, éventuellement suivi d'une phase d'ajustement de faisceaux. Nous renvoyons au chapitre 3 pour un état de l'art et plus de détails sur cette phase. Le résultat de cette étape est un ensemble de n matrices de projection $P_1, \ldots, P_n$. Comme discuté précédemment, l'expression de ces caméras dans une base projective particulière, telle la base canonique avec $P_1 \sim (\ I \quad \mathbf{0})$, n'est pas utile. La solution de reconstruction de caméras proposée par Triggs [216], basée sur les contraintes de fermeture, est attirante. Cet algorithme estime les caméras à partir des tenseurs d'appariement multi-vues : la matrice fondamentale, et les tenseurs trifocaux et quadrifocaux, que l'on estime à partir de correspondances de points.

5.5.1.2 Plans

Dans le cas multi-vues, on ne peut plus paramétrer les plans par leurs équations réduites. On les paramétrise par leurs coordonnées homogènes. La difficulté est de trouver un critère linéaire basé sur des quantités image et

exploitant toutes les données de manière uniforme. Le choix du critère basé sur le transfert par l'homographie de plan induite, utilisé dans le cas de deux vues, est naturel. Nous proposons une extension de ce critère au cas multi-vues.

Toute observation d'un point sur le plan peut être reconstruite en intersectant le rayon de projection qu'elle induit avec le plan. Le point 3D obtenu peut être reprojeté dans les autres images et comparé aux autres observations (aux correspondances) à l'aide d'une distance 2D. Cette mesure d'erreur est illustrée sur la figure 5.10. Notre critère est la somme des distances 2D induites par toutes les observations de chaque point censé être sur

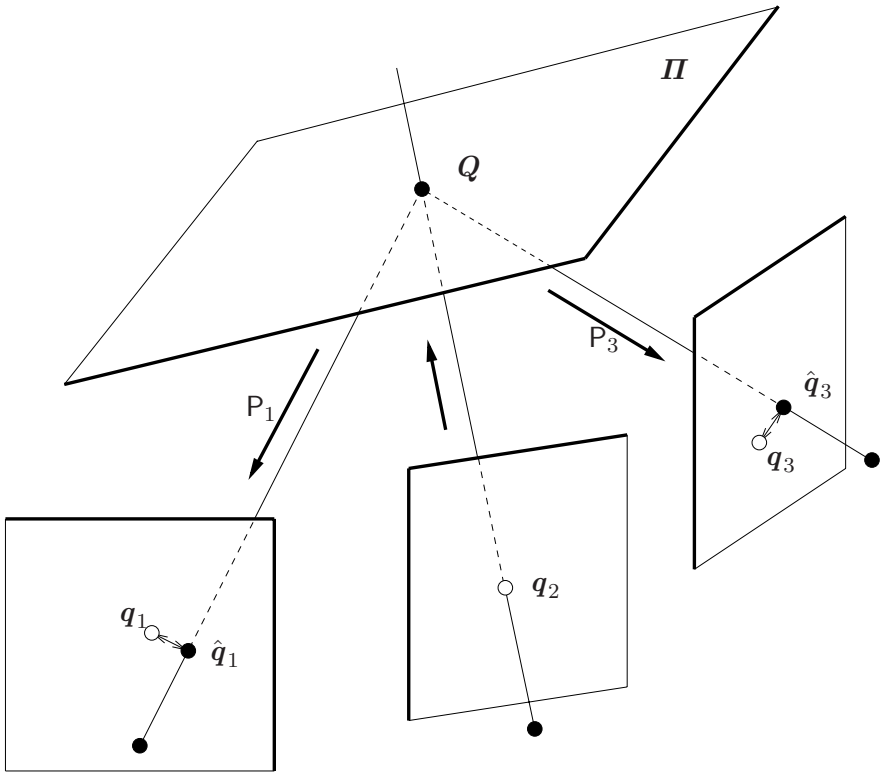


FIG. 5.10 – Illustration de l'erreur minimisée par la méthode linéaire de reconstruction de plans. Chaque point de chaque vue, ici $\boldsymbol{q}_2$, est reconstruit en intersectant son rayon du vue avec le plan $\boldsymbol{\Pi}$ recherché. Le point ainsi reconstruit est reprojeté et comparé avec les correspondances dans les autres vues, ici $\hat{\boldsymbol{q}}_1$ est comparé avec $\boldsymbol{q}_1$ et $\hat{\boldsymbol{q}}_3$ avec $\boldsymbol{q}_3$, à l'aide d'une distance 2D. Une distance algébrique bilinéaire induit un algorithme linéaire, et une distance Euclidienne induit un algorithme quasi-linéaire.

le plan. De manière plus formelle, formulons l'opération d'intersection du rayon de projection issu du point image $\boldsymbol{q}_{ij}$ avec le plan $\boldsymbol{\Pi}$ comme :

$$\mathbf{Q}_j \quad \sim \quad \mathsf{M}_{ij}\boldsymbol{\Pi},$$

où M_{ij} est une matrice (4×4) dont l'expression, étudiée plus loin, dépend de $\mathbf{q}_{ij}$ et de P_i. Notre critère d'erreur est donné par :

$$\mathcal{C}_{\boldsymbol{\Pi}}(\boldsymbol{\Pi}) \quad = \quad \sum_j \sum_i \sum_{i' \neq i} d^2_{A_PP}(\mathbf{q}_{i'j}, \mathsf{P}_{i'}\mathsf{M}_{ij}\boldsymbol{\Pi}).$$

Nous réutilisons la notation $\mathcal{C}_{\boldsymbol{\Pi}}$ pour la fonction de coût car elle se réduit en (5.3), l'erreur de transfert symétrique minimisée par l'estimation d'une homographie de plan par notre méthode donnée en §5.4.1.2, lorsque deux vues sont considérées. Ce critère est minimisé par la solution au sens des moindres carrés du système linéaire suivant :

$$\mathsf{A}_{(t\times 4)}\boldsymbol{\Pi}_{(4\times 1)} \quad = \quad \mathbf{0}_{(t\times 1)}, \tag{5.13}$$

où le nombre de lignes t de A est déterminé par le fait que chaque reprojection fournit 2 équations. Si l'on suppose par exemple que m points sur $\boldsymbol{\Pi}$ sont visibles dans toutes les n vues, on obtient $t = 2nm(n-1)$ (chacun des nm points image induit deux équations pour chaque comparaison avec un de ses $n-1$ correspondants). La matrice A est définie par :

$$\mathsf{A} \quad = \quad \begin{pmatrix} \cdots \\ \mathsf{A}_{ii'j} \\ \cdots \end{pmatrix}, \; i = 1 \ldots n, \; j = 1 \ldots m, \; i' = 1 \ldots n, \; i' \neq i,$$

où la matrice $\mathsf{A}_{ii'j}$ correspond aux équations induites par la reprojection dans la $i^{\text{ème}}$ vue du $j^{\text{ème}}$ point reconstruit à partir de la $i^{\text{ème}}$ vue. Elle est donnée par $\|\mathsf{A}_{ii'j}\boldsymbol{\Pi}\|^2 = d^2_{A_PP}(\mathbf{q}_{i'j}, \mathsf{P}_{i'}\mathsf{M}_{ij}\boldsymbol{\Pi}) = \|\mathsf{S}[\mathbf{q}_{i'j}]_\wedge \mathsf{P}_{i'}\mathsf{M}_{ij}\boldsymbol{\Pi}\|^2$ (voir le chapitre 2, §2.5.2), d'où :

$$(\mathsf{A}_{ii'j})_{(2\times 4)} \quad = \quad \mathsf{S}[\mathbf{q}_{i'j}]_\wedge \mathsf{P}_{i'}\mathsf{M}_{ij}.$$

Intersection d'un rayon de projection avec un plan – Formulation de la matrice M_{ij}. Nous montrons que les coordonnées homogènes $\mathbf{Q}$ du point d'intersection $\boldsymbol{Q}$ d'un rayon de projection issu d'un point $\boldsymbol{q}$ observé par une caméra dont la matrice de projection est P avec un plan $\boldsymbol{\Pi}$ s'écrivent de manière linéaire en $\boldsymbol{\Pi}$, les coordonnées homogènes de $\boldsymbol{\Pi}$. Nous supposons que le point $\boldsymbol{Q}$ est toujours défini de manière unique, c'est à dire que $\boldsymbol{\Pi}$ ne contient pas le centre de projection de la caméra. Les coordonnées homogènes $\mathbf{Q}$ sont données par le vecteur nul d'une matrice (3×4) N modélisant le fait que $\boldsymbol{Q}$ se projette sur $\boldsymbol{q}$ et appartienne à $\boldsymbol{\Pi}$:

$$\underbrace{\begin{pmatrix} \mathsf{S}[\mathbf{q}]_\wedge \mathsf{P} \\ \\ \boldsymbol{\Pi}^\mathsf{T} \end{pmatrix}}_{\mathsf{N}} \mathbf{Q} \quad = \quad \mathbf{0}.$$

Les coefficients du vecteur $\mathbf{Q}$ s'expriment comme une forme trilinéaire en les coefficients des trois lignes de la matrice N. Il existe donc une matrice M de taille (4×4) dépendant de $\mathbf{q}$ et de P et telle que :

$$\mathbf{Q} \quad \sim \quad \mathsf{M}\boldsymbol{\Pi}$$

Définissons la notation suivante pour les lignes de la matrice N :

$$\mathsf{N} \quad = \quad \begin{pmatrix} \bar{\mathbf{N}}_1^\mathsf{T} & N_1 \\ \bar{\mathbf{N}}_2^\mathsf{T} & N_2 \\ \bar{\boldsymbol{\Pi}}^\mathsf{T} & \Pi \end{pmatrix}.$$

En interprétant chaque ligne de N comme l'équation d'un plan, on utilise l'équation (2.53) donnant les coordonnées de l'intersection de trois plans, correspondant au noyau de N, ce qui conduit à :

$$\mathbf{Q} \quad = \quad \begin{pmatrix} \Pi(\bar{\mathbf{N}}_1 \wedge \bar{\mathbf{N}}_2) + N_1(\bar{\mathbf{N}}_2 \wedge \bar{\boldsymbol{\Pi}}) + N_2(\bar{\boldsymbol{\Pi}} \wedge \bar{\mathbf{N}}_1) \\ \\ -(\bar{\mathbf{N}}_1, \bar{\mathbf{N}}_2, \bar{\boldsymbol{\Pi}}) \end{pmatrix},$$

que l'on réorganise pour obtenir l'expression de M :

$$\mathsf{M}_{(4\times 4)} \quad = \quad \begin{pmatrix} N_1[\bar{\mathbf{N}}_2]_\wedge + N_2[\bar{\mathbf{N}}_1]_\wedge & \bar{\mathbf{N}}_1 \wedge \bar{\mathbf{N}}_2 \\ \\ -\bar{\mathbf{N}}_1^\mathsf{T}[\bar{\mathbf{N}}_2]_\wedge & 0 \end{pmatrix}.$$

Cette matrice n'est autre que la matrice de Plücker duale de la droite de vue du point image $\boldsymbol{q}$, voir le chapitre 2, §2.12 ou, par exemple [93, §2.2.2]. Elle affecte à un plan de coordonnées $\boldsymbol{\Pi}$, les coordonnées $\mathbf{Q} \sim \mathsf{M}\boldsymbol{\Pi}$ de l'intersection des plans de coordonnées $\mathbf{N}_1$ et $\mathbf{N}_2$ avec $\boldsymbol{\Pi}$, ou bien de la droite de vue avec $\boldsymbol{\Pi}$.

5.5.1.3 Points non contraints – Triangulation

Les plans étant reconstruits, le but est d'obtenir une reconstruction de points satisfaisant les contraintes de multi-coplanarité. La méthode de triangulation optimale de Hartley et Sturm [85] ne s'étend pas du cas de deux vues vers le cas multi-vues, voir la thèse de Sturm [195, §4.5.5]. Généralisons le critère d'erreur algébrique employé dans le cas de deux vues :

$$\mathcal{C}_{\mathbf{Q}}(\mathbf{Q}) \;=\; \textstyle\sum_i d^2_{A_PP}(\mathbf{q}_i, \mathsf{P}_i\mathbf{Q}) \;=\; \|\mathsf{A}\mathbf{Q}\|^2. \tag{5.14}$$

Ce critère est minimisé par la résolution au sens des moindres carrés du système linéaire suivant :

$$\mathsf{A}\mathbf{Q} \;=\; \mathbf{0} \quad \text{avec} \quad \mathsf{A} \;\sim\; \begin{pmatrix} \cdots \\ \mathsf{S}[\mathbf{q}_i]_\wedge \mathsf{P}_i \\ \cdots \end{pmatrix}. \tag{5.15}$$

Comme précédemment, une résolution quasi-linéaire basée sur le schéma exposé en §2.6.2, chapitre 2, minimise l'erreur de reprojection, ce qui est optimal.

5.5.1.4 Points sur un plan – Triangulation contrainte par coplanarité

Une solution classique à ce problème est de reconstruire les points en ignorant les contraintes géométriques puis de corriger leur position 3D. Cette solution n'est applicable en espace projectif, où la notion de projection orthogonale n'existe pas, qu'en recourant à un critère de correction basé image. Nous proposons une *triangulation contrainte par coplanarité*. Notre solution est basée sur une méthode d'optimisation linéaire sous contrainte linéaire, voir la section 2.6.4.2 du chapitre 2. L'idée est d'exprimer les coordonnées homogènes de $\boldsymbol{Q}$ dans un espace de dimension 3 où la contrainte de coplanarité $\mathbf{\Pi}^\mathsf{T}\mathbf{Q} = 0$ est satisfaite, l'espace nul de $\mathbf{\Pi}^\mathsf{T}$. Une base orthonormale de cet espace est donnée par la décomposition en valeurs singulières suivante :

$$\mathbf{\Pi}^\mathsf{T} \;\sim\; \mathbf{u}^\mathsf{T}\mathrm{diag}(1,0,0,0)(\mathbf{v}_{(4\times1)} \;\; \bar{\mathsf{V}}_{(4\times3)})^\mathsf{T}$$

En effet, $\mathbf{\Pi}^\mathsf{T}\bar{\mathsf{V}} = \mathbf{0}$. Nous exprimons les coordonnées de $\boldsymbol{Q}$ dans cet espace par le vecteur $\boldsymbol{\gamma}$ de taille (3×1) : $\mathbf{Q} = \bar{\mathsf{V}}\boldsymbol{\gamma}$. Remplaçons cette paramétrisation dans l'expression de la fonction de coût (5.14) :

$$\mathcal{C}_{\mathbf{Q}} \;=\; \|\mathsf{A}\mathbf{Q}\|^2 \;=\; \|\mathsf{A}\bar{\mathsf{V}}\boldsymbol{\gamma}\|^2.$$

Le vecteur singulier associé à la plus petite valeur singulière de la matrice $\mathsf{A}\bar{\mathsf{V}}$ donne le vecteur $\boldsymbol{\gamma}$ minimisant le critère d'erreur (5.14), sous la contrainte $\|\boldsymbol{\gamma}\|^2 = 1$. Notons que comme $\bar{\mathsf{V}}$ est orthonormale, $\|\mathbf{Q}\|^2 = \|\bar{\mathsf{V}}\boldsymbol{\gamma}\|^2 = \|\boldsymbol{\gamma}\|^2 = 1$. Une résolution quasi-linéaire permet de minimiser l'erreur de reprojection.

5.5.1.5 Points sur deux plans – Triangulation contrainte par double coplanarité

Nous proposons une *triangulation contrainte par double coplanarité*. Notre solution est basée sur la même idée que pour le cas précédent. Nous calculons une base orthonormale d'un espace de dimension 2 où les deux contraintes de coplanarité sont satisfaites par la décomposition en valeurs singulières suivante :

$$\begin{pmatrix} \mathbf{\Pi}^\mathsf{T} \\ \mathbf{\Pi}'^\mathsf{T} \end{pmatrix} \;\sim\; \mathsf{U}_{(2\times4)}\mathrm{diag}(1,\alpha,0,0)(\mathbf{v}_{(4\times1)} \;\; \mathbf{v}'_{(4\times1)} \;\; \bar{\mathsf{V}}_{(4\times2)})^\mathsf{T}.$$

Comme dans le cas précédent, remplaçons la paramétrisation minimale $\mathbf{Q} \sim \bar{\mathsf{V}}\boldsymbol{\gamma}$, où $\boldsymbol{\gamma}$ est un vecteur (2×1) dans l'expression de la fonction de coût (5.14). On arrive alors à un système linéaire homogène de taille (4×2). Une résolution quasi-linéaire permet de minimiser l'erreur de reprojection.

5.5.1.6 Points sur trois plans – Intersection de trois plans

Un point sur trois plans n'a aucun degré de liberté car il est déterminé par l'intersection des trois plans. L'équation (2.53) donne les coordonnées de l'intersection de trois plans en position générale. On renvoie au même cas traité pour deux vues en §5.4.1.6 pour plus de détails.

5.5.1.7 Autres cas – Points observés dans moins de deux vues

Ces cas sont traités en détails dans le cas de deux vues. Les solutions dans le cas multi-vues sont identiques. On renvoie à la section 5.4.1.7 pour plus de détails.

5.5.2 Ajustement de faisceaux plan par morceaux

La section 5.3.2 décrit les différents systèmes d'ajustement de faisceaux prenant en compte les contraintes de multi-coplanarité. Les quelques approches théoriquement optimales font appel à des techniques d'optimisation non-linéaire coûteuses et compliquées. Nous proposons une méthode basée sur une paramétrisation minimale de la structure, incorporant les contraintes de multi-coplanarité. La minimisation de l'erreur de reprojection peut alors être effectuée sans contrainte non-linéaire, par un algorithme d'optimisation non-linéaire standard. Pour la paramétrisation des caméras, nous renvoyons à la section 3.5.5 du chapitre 3.

Contrairement au cas de deux vues où la même paramétrisation de la structure est utilisée lors de l'initialisation et de l'ajustement de faisceaux, on utilise une paramétrisation différente pour chacune de ces phases dans le cas multi-vues. En effet, lors de l'initialisation, les points sont reconstruits après les plans. Les contraintes de coplanarité du style $\mathbf{\Pi}^\mathsf{T}\mathbf{Q} = 0$ sont linéaires en $\mathbf{Q}$. Lors de l'ajustement de faisceaux, les points et les plans sont estimés simultanément. Les contraintes sont donc bilinéaires.

La méthode proposée est une forme de généralisation de la méthode spécifique au cas de deux vues. Elle est moins directe car l'utilisation d'une base de reconstruction canonique ne permet pas de simplifier le problème. Notre paramétrisation est basée sur l'élimination de coefficients des coordonnées homogènes représentants les points sous contraintes de multi-coplanarité. Pour illustrer la difficulté, nous considérons le cas d'un point 2D $\boldsymbol{q}$ sur une droite 2D $\boldsymbol{l}$, similaire au cas d'un point 3D sur un plan. La contrainte s'exprime algébriquement par $\mathbf{q}^\mathsf{T}\mathbf{l} = 0$. Elle est satisfaite par tout point dont les coordonnées homogènes sont dans l'espace nul bidimensionnel de $\mathbf{l}^\mathsf{T}$. L'approche qui vient à l'idée naturellement est de calculer une base de cet espace nul et d'exprimer les coordonnées du point $\boldsymbol{q}$ dans cette base, comme dans le cas de l'initialisation. Nous examinons deux manières de former une telle base et montrons qu'elles ne sont pas appropriées pour notre problème.

La première solution est donnée en formant la matrice anti-symétrique (3×3) $[\mathbf{l}]_\wedge$ associée au produit vectoriel par le vecteur $\mathbf{l}$. Les colonnes de cette matrices forment une base de l'espace nul recherché car $\mathbf{l}^\mathsf{T}[\mathbf{l}]_\wedge = \mathbf{0}^\mathsf{T}$. Le rang de cette matrice est 2, ce qui correspond au fait que l'espace nul recherché est de dimension 2. Les coordonnées de tout point sur $\boldsymbol{l}$ peuvent donc être représentées par une combinaison linéaire des 3 colonnes de $[\mathbf{l}]_\wedge$, ce qui implique donc 3 coefficients formant un vecteur de coordonnées homogènes. Cette paramétrisation n'est pas consistante car un point sur une droite n'a qu'un degré de liberté. On peut alors penser à n'utiliser que deux colonnes de $[\mathbf{l}]_\wedge$ comme base de l'espace nul, par exemple les deux dernières, $\mathbf{l}_2$ et $\mathbf{l}_3$. Dans ce cas, la représentation est consistante, mais n'est plus complète : le point de coordonnées $\mathbf{l}_1$ est sur la droite $\boldsymbol{l}$ mais ne peut pas être représenté par une combinaison linéaire de $\mathbf{l}_2$ et $\mathbf{l}_3$. Notons qu'il y a d'autres possibilités de base "triviales" pour l'espace nul de $\mathbf{l}^\mathsf{T}$. Elles conduisent toutes au même problème.

La deuxième solution est de calculer une base orthonormale de cet espace nul en utilisant par exemple la décomposition en valeurs singulières suivante :

$$\mathbf{l}^\mathsf{T} \quad \sim \quad \mathbf{l}^\mathsf{T}\mathrm{diag}(1,0,0)\begin{pmatrix}\mathbf{l}_{(3\times1)} & \bar{\mathsf{V}}_{(3\times2)}\end{pmatrix}.$$

Dans ce cas, la base donnée par les deux colonnes de $\bar{\mathsf{V}}$ est minimale et la paramétrisation correspondante serait consistante et complète. Le problème est qu'il n'existe pas d'expression analytique simple donnant les coefficient de la matrice $\bar{\mathsf{V}}$ en termes des coefficients de $\mathbf{l}$. Ceci entraînerait que la différenciation analytique de la fonction de paramétrisation prendrait une forme compliquée, voir le travail de Papadopoulo et Lourakis [158]. De plus, cette base devrait être recalculée à chaque étape de la minimisation. L'utilisation d'une décomposition en valeurs singulières n'est pas raisonnable car elle possède des singularités, plus précisément des ambiguïtés discrètes, pouvant conduire à de gros changements numériques de $\bar{\mathsf{V}}$ pour de petits changements de $\mathbf{l}$.

La consistance et la différenciation analytique sont parmi les principales raisons des paramétrisations spécifiques proposées ci-dessous.

5.5.2.1 Plans

Tous les plans modélisés ont 3 degrés de liberté, car les contraintes de multi-coplanarité ne s'exercent que sur les points, grâce à l'hypothèse qu'un point est sur 3 plans au maximum. Un vecteur homogène de taille (4×1) constitue donc une représentation consistante d'un plan. Notons que l'équation réduite utilisée dans le cas de deux vues n'est plus utilisable. En effet, l'équation réduite n'existe que pour les plans ne contenant pas le centre de projection de la caméra de référence (de la caméra dont la matrice de projection est $(\ \mathrm{I}\quad \mathbf{0})$). Or, dans le cas multi-vues, il est possible qu'un plan soit reconstructible et contienne ce centre de projection.

5.5.2.2 Points non contraints

Un point non contraint a 3 degrés de liberté. Comme dans le cas des plans, un vecteur homogène de taille (4×1) en est une représentation consistante. Notons que la paramétrisation d'Hartley et Zisserman [93], voir le chapitre 2, §2.10.1.3, n'est applicable qu'aux points visibles dans la caméra de référence. Nous ne l'utilisons donc pas.

5.5.2.3 Points sur un plan

Soit $\boldsymbol{Q}$ un point contraint sur un plan $\boldsymbol{\Pi}$. Un tel point a 2 degrés de liberté et notre but est de l'exprimer par un vecteur de coordonnées homogènes de taille (3×1), au lieu de la taille (4×1) habituelle, en incorporant la contrainte de coplanarité. Algébriquement, cette contrainte s'écrit $\boldsymbol{\Pi}^{\mathsf{T}}\mathbf{Q} = 0$. Nous cherchons un changement de base projective où les coordonnées de chaque point sur $\boldsymbol{\Pi}$ ont un coefficient fixé à une valeur constante. Ce coefficient pourra ensuite être éliminé de la paramétrisation. Si une telle transformation se conçoit assez intuitivement en espace métrique ou Euclidien, elle est moins directe en espace projectif. Définissons la famille d'homographies 3D $\mathsf{H}^r_{\boldsymbol{\Pi}}$ avec $r \in \{1 \dots 4\}$ comme la matrice identité de taille (4×4) où la $r^{\text{ème}}$ ligne est remplacée par le vecteur-4 $\boldsymbol{\Pi}^{\mathsf{T}}$. Par exemple :

$$\mathsf{H}^1_{\boldsymbol{\Pi}} \sim \begin{pmatrix} & \boldsymbol{\Pi}^{\mathsf{T}} & \\ & & \\ \mathbf{0}_{(3\times1)} & & \mathrm{I}_{(3\times3)} \end{pmatrix}.$$

Cette homographie 3D définit une nouvelle base projective par rapport à la base de reconstruction. Soient $\boldsymbol{\Xi} \sim \mathsf{H}^r_{\boldsymbol{\Pi}}\mathbf{Q}$ les coordonnées de $\boldsymbol{Q}$ dans cette nouvelle base. Par définition de $\mathsf{H}^r_{\boldsymbol{\Pi}}$, si $\boldsymbol{Q} \in \boldsymbol{\Pi}$, nous obtenons $\Xi_r = 0$ et le point $\boldsymbol{Q}$ peut donc être paramétré par $\boldsymbol{\Xi}_{/r}$, le vecteur-3 homogène formé par les 3 coefficients de $\boldsymbol{\Xi}$ d'indice différent de r. Les coordonnées $\mathbf{Q}$ sont ensuite retrouvées par $\mathbf{Q} \sim (\mathsf{H}^r_{\boldsymbol{\Pi}})^{-1}\boldsymbol{\Xi}$.

Il y a 4 possibilités pour le choix de l'indice r. Comme $(\mathsf{H}^r_{\boldsymbol{\Pi}})^{-1}$ est nécessaire pour retrouver $\mathbf{Q}$ à partir de $\boldsymbol{\Xi}$, nous choisissons l'indice r qui maximise la valeur absolue du déterminant de $\mathsf{H}^r_{\boldsymbol{\Pi}}$: $r = \arg\max_r |\det(\mathsf{H}^r_{\boldsymbol{\Pi}})|$, qui conduit à $r = \arg\max_r |\Pi_r|$. Un tel choix assure que $\mathsf{H}^r_{\boldsymbol{\Pi}}$ est une transformation bijective car $\det(\mathsf{H}^r_{\boldsymbol{\Pi}}) = \Pi_r$ est par construction toujours non nul. En effet, $\boldsymbol{\Pi}$ est un vecteur de coordonnées homogènes et a toujours un coefficient non nul.

L'indice r est une sorte de carte de paramétrisation, indiquant quel est le coefficient de $\boldsymbol{\Pi}$ que l'on peut laisser invariant. On suppose que les variations locales de $\boldsymbol{\Pi}$ (lors d'une itération) ne changent pas r. Une paramétrisation minimale modélise la façon dont les points bougent lorsque le plan est changé. Un tel mouvement est difficile à imaginer, surtout en espace projectif.

Le tableau 5.3 résume cette paramétrisation de manière pratique. L'indice du coefficient de $\mathbf{Q}$ éliminé dépend de la valeur numérique de $\boldsymbol{\Pi}$. Il peut donc changer entre deux itérations du processus d'optimisation. Cependant, cela ne crée pas de discontinuité car après chaque itération, les coordonnées sont exprimées dans la base projective originale. Le point est alors reparamétré pour la prochaine itération, et l'indice du coefficient éliminé change sans poser de problème. La paramétrisation est donc utilisée d'une manière locale, ce qui est important pour garder un espace de paramètres lisse et éviter d'y créer des discontinuités.

Notons qu'en espace affine, métrique et Euclidien, le vecteur de coordonnées $\mathbf{Q}$ peut être mis à l'échelle tel que $Q_4 = 1$. Le $4^{\text{ème}}$ coefficient est donc déjà fixé et nous choisissons l'indice r du coefficient à éliminer comme $r \in \{1, 2, 3\}$.

Soit $\boldsymbol{Q}$ un point sous contrainte de coplanarité avec le plan $\boldsymbol{\Pi}$. Le vecteur-4 $\mathbf{Q}$ donne les coordonnées homogènes de $\boldsymbol{Q}$ dans la base projective de reconstruction et le vecteur-3 $\tilde{\mathbf{Q}}$ est une paramétrisation de $\boldsymbol{Q}$ incorporant la contrainte de coplanarité. *Paramétrisation* $(\mathbf{Q} \rightarrow \tilde{\mathbf{Q}})$ *:* $\triangleright$ choisir j tel que $j = argmax_i \lvert\Pi_i\rvert$ sous la contrainte $j \in \{1 \dots 4\}$ dans le cas projectif et $j \in \{1 \dots 3\}$ dans le cas affine, métrique ou Euclidien ; $\triangleright$ $\tilde{\mathbf{Q}} \sim \mathbf{Q}_{/j}$. *Recomposition* $(\tilde{\mathbf{Q}} \rightarrow \mathbf{Q})$ *:* $\triangleright$ calculer $\alpha = -\frac{\sum_{i \neq j} \Pi_i Q_i}{\Pi_j}$; $\triangleright$ $\mathbf{Q} \sim \tilde{\mathbf{Q}}_{j \leftarrow \alpha}$.

TAB. 5.3 – Paramétrisation consistante d'un point sur un plan par élimination.

5.5.2.4 Points sur deux plans

Soit $\boldsymbol{Q}$ un point contraint sur deux plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi'}$. Un tel point a 1 degré de liberté si les deux plans sont différents, et notre but est de le représenter par un vecteur de coordonnées homogènes de taille (2×1), au lieu du vecteur de taille (4×1) standard, en y incorporant les deux contraintes de coplanarité.

Nous suivons un raisonnement similaire au cas précédent. Nous définissons la famille d'homographies 3D $\mathsf{H}^{r,r'}_{\boldsymbol{\Pi},\boldsymbol{\Pi}'}$ comme les matrices $\mathsf{H}^{r}_{\boldsymbol{\Pi}}$ où la $r'^{\text{ème}}$ ligne est remplacée par le vecteur-4 $\boldsymbol{\Pi}'^{\mathsf{T}}$. Par exemple :

$$\mathsf{H}^{1,2}_{\boldsymbol{\Pi},\boldsymbol{\Pi}'} \sim \begin{pmatrix} \boldsymbol{\Pi}^{\mathsf{T}} \\ \boldsymbol{\Pi}'^{\mathsf{T}} \\ 0_{(2\times 2)} \quad \mathrm{I}_{(2\times 2)} \end{pmatrix}.$$

On pose $\boldsymbol{\Xi} \sim \mathsf{H}^{r,r'}_{\boldsymbol{\Pi},\boldsymbol{\Pi}'}\mathbf{Q}$. Par définition de $\mathsf{H}^{r,r'}_{\boldsymbol{\Pi},\boldsymbol{\Pi}'}$, on obtient $\Xi_r = \Xi_{r'} = 0$ et le point $\boldsymbol{Q}$ peut être paramétré par le vecteur $\boldsymbol{\Xi}_{/r,r'}$, c'est à dire le vecteur de coordonnées homogènes de taille (2×1) formé par les deux coefficients de $\mathbf{Q}$ d'indice différent de r et de r'. On retrouve ensuite $\mathbf{Q}$ par $\mathbf{Q} \sim \left(\mathsf{H}^{r,r'}_{\boldsymbol{\Pi},\boldsymbol{\Pi}'}\right)^{-1}\boldsymbol{\Xi}$.

Comme r et r' doivent être différents, cela laisse $4 \times 3 = 12$ choix possibles. Comme $\left(\mathsf{H}^{r,r'}_{\boldsymbol{\Pi},\boldsymbol{\Pi}'}\right)^{-1}$ est nécessaire, nous choisissons r et r' de telle sorte que le déterminant de $\mathsf{H}^{r,r'}_{\boldsymbol{\Pi},\boldsymbol{\Pi}'}$ soit maximisé, en valeur absolue. La formulation d'un algorithme pratique à partir de cette paramétrisation est très similaire au cas précédent. Si une reconstruction affine, métrique ou Euclidienne est considérée, les indices r et r' sont choisis parmi $\{1, 2, 3\}$.

5.5.2.5 Points sur trois plans

Soit $\boldsymbol{Q}$ un point contraint sur les plans $\boldsymbol{\Pi}$, $\boldsymbol{\Pi'}$ et $\boldsymbol{\Pi''}$. Comme mentionné précédemment, il est trivial de voir qu'un point sur trois plans en position générale, c'est à dire différents et ne formant pas un faisceau, n'a aucun degré de liberté, car il est déterminé par l'intersection des trois plans. De tels points ne sont donc pas représentés dans la paramétrisation et leurs coordonnées sont données directement en termes des trois équations de plan. On renvoie à la section 5.4.1.6 pour plus de détails.

5.5.2.6 Paramétrisation des vecteurs de coordonnées homogènes

La paramétrisation décrite ci-dessus est consistante dans le sens où elle incorpore les contraintes géométriques, mais elle n'est pas minimale dans le sens où les vecteurs de paramètres représentant chaque plan ou point sont définis à un facteur près. Lors de l'ajustement de faisceaux, nous sélectionnons à chaque itération le coefficient de valeur absolue maximale pour chaque vecteur de coordonnées homogènes et ne conduisons l'optimisation que sur les autres coefficients. L'optimisation est donc localement minimale.

5.5.2.7 Structure et résolution des équations normales

Les équations normales, dont la résolution est nécessaire à chaque étape de l'ajustement de faisceaux, ont une structure éparse par bloque, grâce à notre paramétrisation minimale et à l'hypothèse qu'un point ne soit pas sur plus de trois plans. La structure de ces équations est identique (pour les plans et les points) à celle montrée dans le cas de deux vues. La méthode de résolution est la même. Nous renvoyons à la section 5.4.2.2 pour plus de détails.

5.6 Autres types de contraintes

Nous traitons la modélisation algébrique des contraintes de multi-coplanarité entre un point et plus de trois plans. Nous examinons ensuite la formulation et l'incorporation de contraintes entre les plans modélisés.

Contraintes de multi-coplanarité, points sur plus de trois plans. Comme mentionné précédemment, il est rare d'avoir à modéliser de telles contraintes. Des points sur quatre plans peuvent être rencontrés, mais plus le nombre de plans augmente, plus il est rare de rencontrer de tels cas. Gérer tous ces cas ajouterait une grande complexité au système, dans le sens où les contraintes ne pourraient plus être exprimées seulement sur les points, mais devraient l'être aussi sur les plans, ce qui pourrait créer un graphe de contraintes avec des redondances et des cycles. Nous examinons dans ce paragraphe comment ces contraintes pourraient être manipulées de manière algébrique, dans le cas de quatre plans $\boldsymbol{\Pi}$, $\boldsymbol{\Pi}'$, $\boldsymbol{\Pi}''$ et $\boldsymbol{\Pi}'''$. Les cas d'ordre supérieur, bien que plus compliqués pourraient être gérés de manière similaire. Les contraintes sont exprimées algébriquement par l'équation $\mathsf{B}\mathbf{Q} = \mathbf{0}$, où B est une matrice (4×4) donnée par :

$$\mathsf{B}^{\mathsf{T}} \quad \sim \quad \begin{pmatrix} \mathbf{\Pi} & \mathbf{\Pi}' & \mathbf{\Pi}'' & \mathbf{\Pi}''' \end{pmatrix}.$$

Cette équation implique que la matrice B a un espace nul de dimension au moins 1, c'est à dire $\det(\mathsf{B}) = 0$, ce qui donne une contrainte quadrilinéaire sur les coefficients des équations de plan. Si on choisit par exemple de contraindre la position du plan $\boldsymbol{\Pi}$, alors un de ses coefficients peut être supprimé de la paramétrisation en appliquant un schéma similaire au cas de la paramétrisation d'un point sur un plan.

Contraintes entre plans. Les contraintes intéressantes entre plans sont presque toutes de nature affine ou métrique. Dans ce paragraphe, nous donnons une esquisse de la modélisation algébrique de telles contraintes. Nous nous concentrons sur les cas des contraintes d'orthogonalité et de parallélisme de deux plans. Un traitement plus complet de ce type de contraintes est hors du cadre de ce chapitre.

Considérons deux plans $\boldsymbol{\Pi}$et $\boldsymbol{\Pi}'$ contraints d'être perpendiculaires. Cette contrainte peut être exprimée algébriquement en considérant que le produit scalaire entre les coordonnées des vecteurs normaux des deux plans s'annule :

$$\Pi_1\Pi'_1 + \Pi_2\Pi'_2 + \Pi_3\Pi'_3 \quad = \quad 0.$$

Cette contrainte est bilinéaire. On peut utiliser une stratégie d'élimination d'un coefficient pour contraindre la position d'un des deux plans. Ceci conduit au même problème que celui de la modélisation de la coplanarité entre un point et un plan.

Le parallélisme de deux plans peut être formulé algébriquement en considérant que les vecteurs normaux de ces deux plans doivent être égaux, à l'échelle près, ce qui est équivalent à annuler leur produit vectoriel :

$$\begin{cases} \Pi_2\Pi'_3 - \Pi_3\Pi'_2 & = & 0 \\ \Pi_3\Pi'_1 - \Pi_1\Pi'_3 & = & 0 \\ \Pi_1\Pi'_2 - \Pi_2\Pi'_1 & = & 0. \end{cases}$$

Parmi ces trois équations, seules deux sont indépendantes, mais on ne peut pas en choisir deux a priori. Donc, selon quel plan doit être contraint, et sur quel axe, deux équations sont utilisées pour éliminer deux coefficients des coordonnées homogènes du plan. Ces équations étant bilinéaires, on retombe sur le problème de la modélisation de la multi-coplanarité d'un point avec deux plans.

5.7 Résultats expérimentaux

Cette section a deux buts principaux. Le premier est de comparer nos méthodes à d'autres approches. Le deuxième est d'estimer les limites de validité de l'utilisation de contraintes de multi-coplanarité pour la reconstruction de scènes réelles, c'est à dire lorsque les points ne sont pas exactement sur les plans. En d'autres termes, nous cherchons à évaluer dans quelles conditions l'utilisation de contraintes de multi-coplanarité améliore la reconstruction, en dépit du caractère approximatif de ces contraintes sur une scène réelle. Nous utilisons des données simulées puis des données réelles. Nous détaillons ci-dessous les méthodes comparées et le critère de mesure de la précision d'une reconstruction.

Méthodes comparées. Nous comparons les méthodes d'ajustement de faisceaux suivantes :

- ▷ POINT_OPTIMAL est un ajustement de faisceaux standard, ignorant les contraintes géométriques et qui sert de référence pour l'évaluation des limites de l'utilisation de ces contraintes. Nous renvoyons au chapitre 3 pour plus de détails sur l'ajustement de faisceaux ;
- ▷ PLAN_POIDS est la méthode de Szeliski et Torr [200], modélisant les contraintes géométriques en utilisant des équations fortement pondérées ajoutées à l'erreur de reprojection. La structure estimée n'est qu'approximativement plane par morceaux. Nous renvoyons à la section 5.3.2 pour plus de détails ;
- ▷ PLAN_OPTIMAL utilise les paramétrisations décrites dans ce chapitre pour modéliser explicitement et exactement les contraintes de multi-coplanarité. Notons que la paramétrisation spécifique au cas de deux vues donne les mêmes résultats que celle concernant le cas multi-vues ;
- ▷ PLAN_HOMOGRAPHIE utilise des homographies planes estimées au maximum de vraisemblance pour chaque groupe de points coplanaires. Elle est proche de la méthode de Xu et al. [231]. Cette méthode n'est applicable que lorsque deux vues sont utilisées. Elle n'exploite que les contraintes de coplanarité simple, c'est à dire entre un point et un plan. Une fois les homographies estimées, les points sont corrigés pour satisfaire exactement ces homographies, et une procédure de reconstruction standard non contrainte basée sur ces points est lancée. Nous renvoyons au chapitre 4 pour plus de détails sur la reconstruction à partir de deux vues.

Les différentes méthodes d'initialisation entraînent en général la convergence des ajustements de faisceaux vers les mêmes minima locaux. Toutefois, le nombre d'itérations effectuées par ces ajustements de faisceaux pour atteindre la solution peut être assez différent selon l'initialisation.

Mesure de la qualité de la reconstruction. Nous mesurons la qualité d'une reconstruction en utilisant une erreur résiduelle 3D. L'erreur 3D est plus importante que l'erreur de reprojection. En effet, l'erreur de reprojection ne reflète que très indirectement la qualité de la reconstruction. En particulier, lorsque des méthodes utilisant des contraintes géométriques sont mises en œuvre, l'erreur de reprojection pour ces méthodes est toujours plus élevée que pour une méthode sans contrainte, alors que la précision de la reconstruction est souvent meilleure.

Pour calculer cette erreur 3D, nous estimons la meilleure homographie 3D H entre la reconstruction estimée, les points $\boldsymbol{Q}_j$, et la structure Euclidienne simulée $\underline{\boldsymbol{Q}}_j$, en minimisant le critère suivant :

$$E \;=\; \sqrt{\frac{1}{m}\sum_{j=1}^{m} d_{PP}^2(\mathsf{H}\mathbf{Q}_j, \underline{\mathbf{Q}}_j)},$$

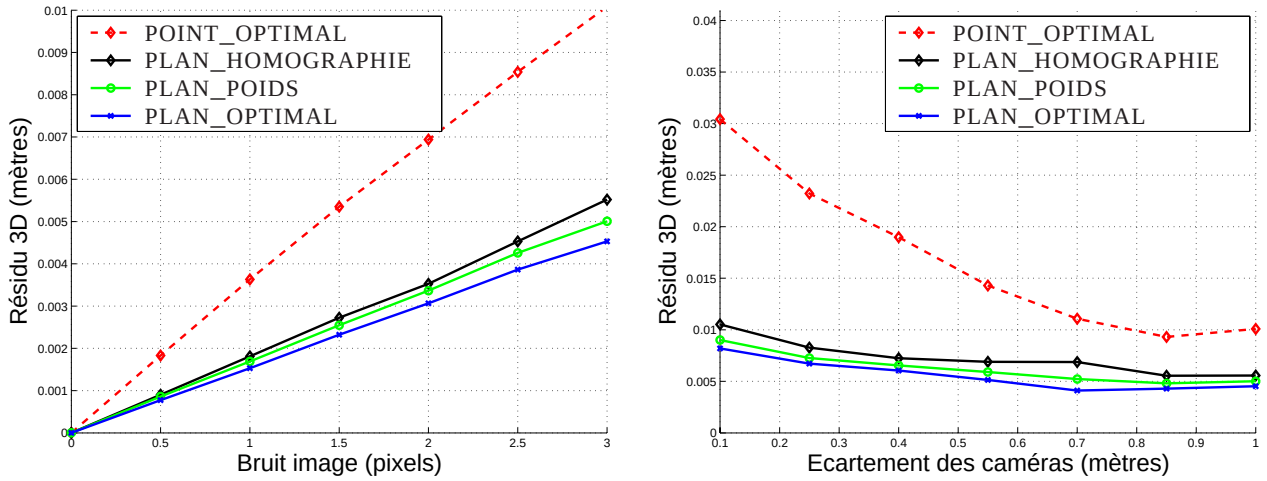
où d_{PP} est la distance Euclidienne entre deux points. Nous renvoyons aux travaux de Csurka et al. [47] pour plus de détails sur le calcul d'une telle homographie. La valeur de E est notre mesure de qualité. Nous mesurons des valeurs médianes sur 100 simulations. Une erreur de même type est aussi calculée pour évaluer la qualité des positions des caméras reconstruites, voir ci-après.

5.7.1 Données simulées

Scène simulée. La scène simulée consiste en un cube observé par un jeu de caméras. Des points sont générés sur le cube et peuvent être écartés des plans les contenant afin de simuler une planarité imparfaite. Ces points

sont projetés dans les images, où un bruit Gaussien est ajouté. Les paramètres par défaut de ce scénario sont les suivants. Jusqu'à 50, 10 et 1 points sont générés respectivement sur chaque face, arête et sommet du cube. Deux caméras avec une distance focale de 1000 pixels et un écartement de 1 mètre sont situées à 10 mètres du cube. La déviation standard du bruit image peut aller jusqu'à 3 pixels. Les paramètres intrinsèques des caméras ne sont pas utilisés, les reconstructions obtenues sont donc projectives. Nous varions indépendamment différents paramètres de cette scène pour comparer les approches dans des conditions variées, et notamment lorsque la planarité par morceaux n'est pas exacte.

Planarité par morceaux exacte. Les figures 5.11, 5.12 et 5.13 montrent les résultats obtenus lorsque différents paramètres d'observation sont modifiés. Sur les figures 5.11 (a) et (b) respectivement, l'écart-type du bruit image est varié entre 0 et 3 pixels et l'écartement des caméras est varié entre 0,1 et 1 mètres. Sur la figure 5.12 (a), le nombre de points est égal à respectivement 50α, 10α et 1 pour chaque face, arête et sommet du cube simulé, et α est varié entre 0,1 et 1. Sur la figure 5.12 (b), le nombre de vues est varié de 2 à 10. Les différentes caméras sont situées telles que l'écartement entre deux caméras consécutives soit de 1 mètre. Sur la figure 5.13 (a), la distance entre le cube et les caméras est variée entre 10 et 20 mètres.

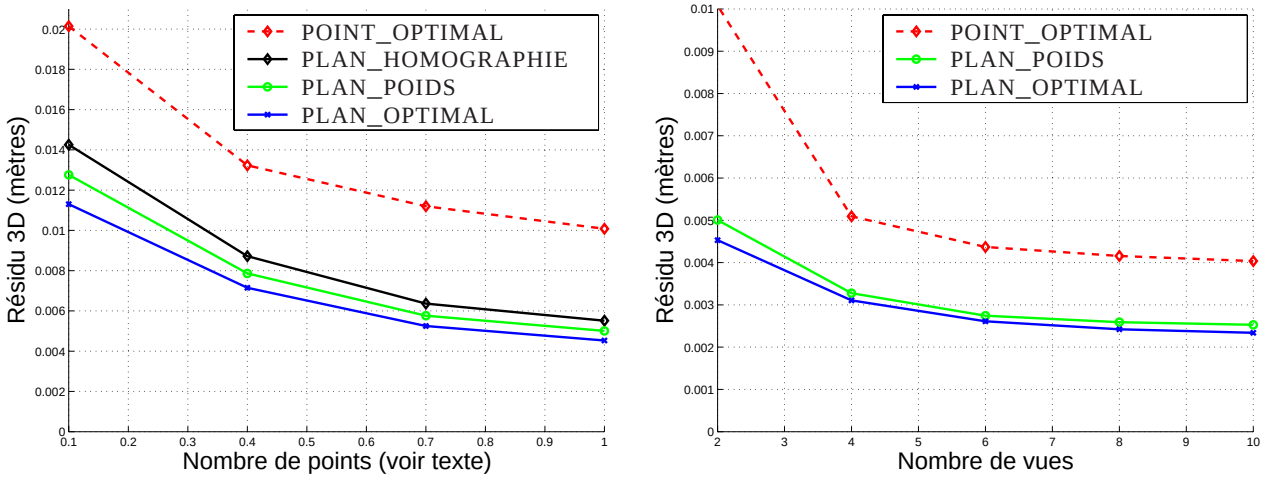


(a) (b)

FIG. 5.11 – Comparaison de l'erreur résiduelle 3D pour différentes approches pour différents niveaux de bruit image (a) et écartements entre les caméras (b).

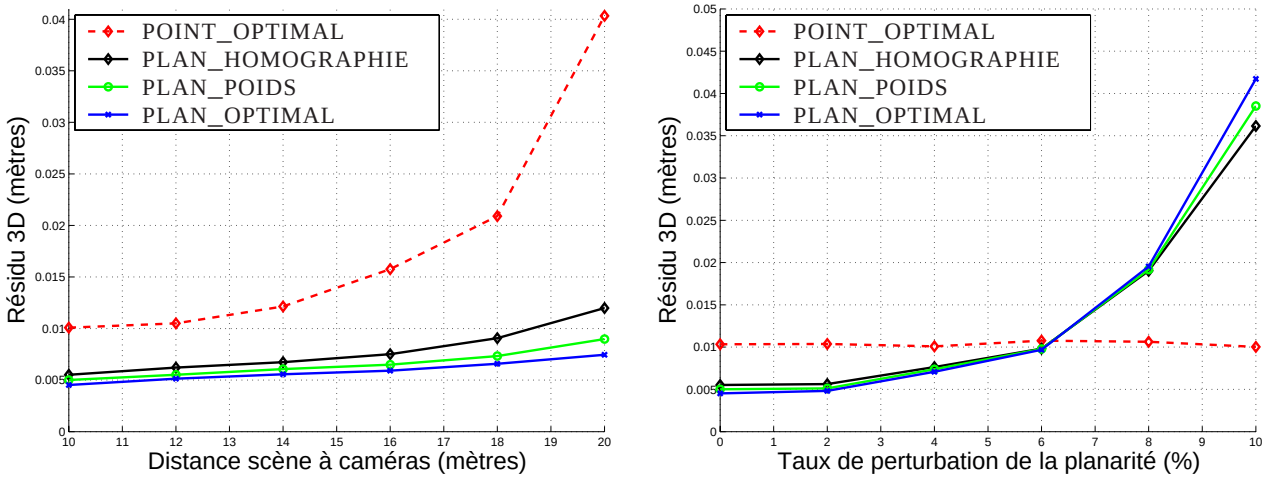
Dans tous les cas, la méthode POINT_OPTIMAL basée sur une reconstruction individuelle de chaque point donne des résultats d'une qualité inférieure aux méthodes PLAN_* utilisant les contraintes de multi-coplanarité (l'erreur 3D est au moins deux fois moindre pour ces méthodes). La méthode PLAN_OPTIMAL donne des résultats légèrement meilleurs que PLAN_POIDS dans tous les cas. Finalement, la méthode PLAN_HOMOGRAPHIE donne des résultats légèrement moins bons que PLAN_POIDS.

Un aspect non montré sur ces graphes, dû à l'utilisation d'une valeur médiane sur un grand nombre de tirages, est que les méthodes POINT_OPTIMAL et PLAN_POIDS ont un pourcentage de convergence inférieur à PLAN_OPTIMAL et PLAN_HOMOGRAPHIE, spécialement lorsque la configuration géométrique est instable (bruit image élevé, petit écartement entre les caméras, grande distance scène à caméras, etc.). Par convergence, nous entendons amélioration de l'erreur de reprojection donnée par l'estimation initiale. Par exemple, les pourcentages d'estimations convergentes sur 1000 tirages sont 95,2%, 89,1%, 97,5% et 97,3% pour POINT_OPTIMAL, PLAN_POIDS, PLAN_OPTIMAL et PLAN_HOMOGRAPHIE respectivement, pour une distance scène à caméras de 20 mètres et un écartement entre caméras de 0,1 mètres. Une explication de ce phénomène est que les configurations instables testées sont souvent induites par le fait que les points observés apparaissent très regroupés dans les images. La reconstruction est de mauvaise qualité, mais l'erreur de reprojection, qui est le critère de minimisation, est petite. La solution initiale est donc souvent bonne dans le sens où elle correspond



(a) (b)

FIG. 5.12 – Comparaison de l'erreur résiduelle 3D pour différentes approches pour différents nombres de points (a) et nombres de vues (b). Le nombre de points est 50α, 10α et 1 sur chaque face, arête et sommet du cube simulé, α étant indiqué sur l'axe des abscisses de (a). La méthode PLAN_HOMOGRAPHIE n'apparait pas en (b) car elle n'est utilisable que lorsque deux vues sont considérées.



(a) (b)

FIG. 5.13 – Comparaison de l'erreur résiduelle 3D pour différentes approches pour différentes distances scène à caméras (a) et différents taux de perturbation de la planarité (b).

$n = 2$	3 m.	10 m.	20 m.
1 pixel	0.5%	2%	4%
3 pixels	2%	6%	9%

$n = 10$	3 m.	10 m.	20 m.
1 pixel	0.3%	1.2%	3%
3 pixels	1.3%	4%	8%

TAB. 5.4 – Seuil de rupture ε déterminé expérimentalement pour différentes configurations (bruit image, nombre de vues n et distances scènes à caméras).

déjà à un minimum local de l'erreur de reprojection.

Planarité par morceaux approchée. Les résultats précédents sont en faveur des approches utilisant les contraintes de multi-coplanarité. Cependant, il faut prendre en compte le fait que la scène simulée est exactement plane par morceaux, ce qui n'est pas le cas lorsque des données réelles sont utilisées, car les surfaces exactement planes n'existent pas dans le monde réel. Pour cette raison, nous examinons jusqu'à quelle déviation de la planarité, l'incorporation des contraintes de multi-coplanarité continue à donner de meilleurs résultats que l'approche non contrainte. Nous perturbons la coplanarité des points en les écartant des plans les contenant par un bruit 3D ajouté dans une direction perpendiculaire aux plans. Intuitivement, on peut prédire qu'à partir d'un certain seuil de perturbation de la planarité, les contraintes de multi-coplanarité ne seront plus valides, et l'ajustement de faisceaux non contraint donnera de meilleurs résultats que les méthodes utilisant les contraintes. Nous définissons le *seuil de rupture*, noté ε, comme le rapport entre le niveau de bruit 3D et la surface observée, au dessus duquel l'ordre des performances s'inverse. La figure 5.13 (b) montre le résultat pour un bruit 3D basé sur une distribution normale avec une déviation standard variant entre 0 et 0,1 mètres, ce qui correspond à 10% de la taille de la scène. Nous observons sur cette figure que $\varepsilon = 6\%$ pour les paramètres par défaut de la scène. Notons que le seuil de rupture dépend de tous les paramètres d'observation de la scène. Nous déterminons expérimentalement le seuil de rupture pour deux valeurs différentes de bruit image, et différentes distances scène à caméras et nombres de vues. Ces valeurs sont données dans le tableau 5.4.

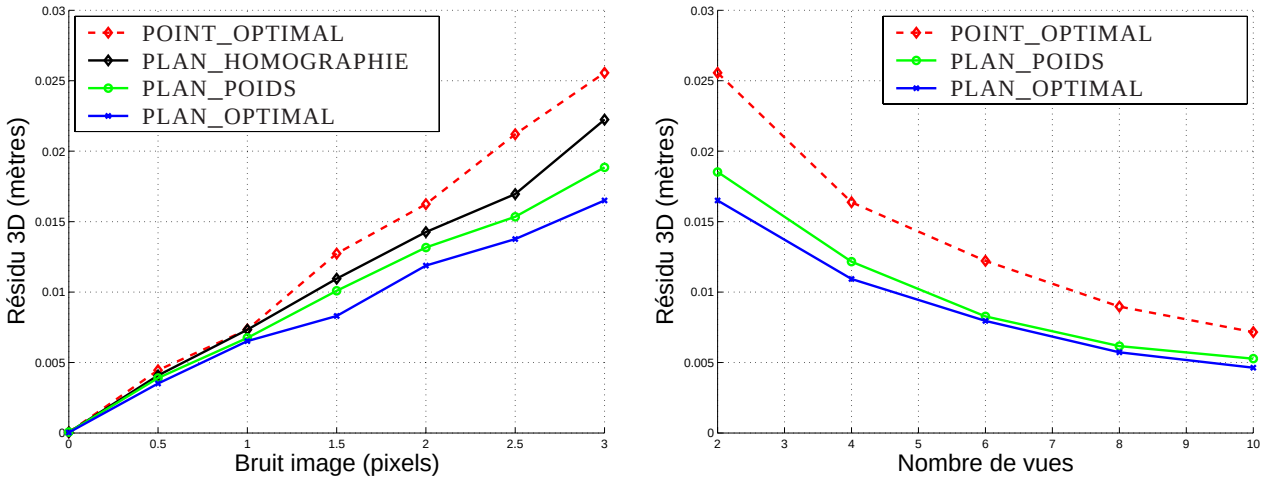
Comme attendu, ces résultats révèlent que plus la configuration est stable, moins l'incorporation des contraintes géométriques est importante. Par exemple, pour une scène située à 3 mètres de 2 caméras et avec 1 pixel de bruit image, seules les surfaces dont le taux de perturbation de la planarité est inférieur à 0,5% gagnent à être modélisées par un plan. Une telle configuration est représentative de la modélisation d'une scène d'intérieur. Lors de la modélisation d'une scène d'extérieur, par exemple pour la reconstruction d'un immeuble, le scénario correspondra plutôt à une distance scène à caméras de 10 mètres. Dans ce cas, on remarque que même les surfaces subissant de grosses perturbations de la planarité ont intérêt à être modélisées par des plans.

Pour résumer, le tableau 5.4 montre que moins la configuration est stable, plus larges sont les perturbations de la planarité tolérées, c'est à dire pour lesquelles l'incorporation des contraintes de multi-coplanarité améliore la qualité de la reconstruction par rapport à un ajustement de faisceaux non contraint. Les valeurs de 1 à plusieurs pourcents du tableau 5.4 représentent des variations relativement larges, supérieures à une grande majorité de surfaces de scènes réelles. En conséquence, nous pouvons conclure qu'il existe de nombreux cas où l'utilisation des contraintes de multi-coplanarité améliore la reconstruction.

Nous avons effectué les mêmes tests en utilisant le calibrage des caméras, c'est à dire avec des reconstructions métriques. Les résultats ne sont pas changés de manière significative, ce qui s'explique par le fait que l'erreur de reprojection est invariante aux transformations projectives de l'espace, et donc au niveau de calibrage utilisé.

Qualité de la position des caméras. La structure reconstruite bénéficie de la modélisation des contraintes de multi-coplanarité. Est-ce le cas pour les caméras reconstruites ? Nous comparons les résultats fournis par différentes méthodes sur la position des caméras. Nous utilisons le scénario décrit précédemment. La qualité de la position des caméras est estimée comme suit. Nous extrayons les n centres de projection $\boldsymbol{C}_i$ des caméras reconstruites et calculons l'erreur résiduelle 3D, comme une fonction des distances avec les centres de projection simulés $\underline{\boldsymbol{C}}_i$ en espace Euclidien. Cette erreur est donnée par la fonction $E_{\text{caméras}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} d_{PP}^2(\mathsf{H}\mathbf{C}_i, \underline{\mathbf{C}}_i)}$.

L'homographie 3D H est estimée en minimisant le critère sur la structure E, c'est à dire en minimisant les distances entre les points reconstruits et les points simulés, car une estimation basée sur les centres de projection seulement serait très sensible au bruit à cause du faible nombre de données. Nous mesurons la valeur médiane de $E_{\text{caméras}}$ sur 100 simulations.



(a) (b)

FIG. 5.14 – Comparaison de l'erreur résiduelle 3D sur la position des caméras pour différentes approches et différents niveaux de bruit image (a) et différents nombres de vues (b). Notons que la méthode PLAN_HOMOGRAPHIE n'est pas visible sur (b) car elle n'est applicable qu'à deux vues.

La figure 5.14 (a) montre le résultat lorsque le bruit image est varié de 0 à 3 pixels et la figure 5.14 (b) lorsque le nombre de vues est varié de 2 à 10, avec un bruit image de 3 pixels. On retrouve des résultats semblables à ceux observés précédemment sur la structure. La méthode POINT_OPTIMAL n'utilisant pas l'information de multi-coplanarité donne de moins bons résultats que les autres méthodes. La méthode PLAN_OPTIMAL donne les meilleurs résultats, suivie par la méthode PLAN_POIDS et finalement la méthode PLAN_HOMOGRAPHIE. Nous observons que la différence entre les méthodes PLAN_* basées sur les contraintes de multi-coplanarité et la méthode POINT_OPTIMAL est réduite comparée aux résultats sur la structure. Nous observons aussi que dans tous les cas, l'erreur est plus élevée que sur la structure. Ceci est dû à deux phénomènes. Le premier est que l'homographie 3D H intervenant dans l'expression de l'erreur est calculée de manière à minimiser l'erreur sur la structure. Le deuxième est que les contraintes de multi-coplanarité concernent la structure et n'améliore qu'indirectement l'estimation des caméras.

5.7.2 Données réelles

La mise en œuvre de la reconstruction plane par morceaux sur des images réelles requiert la mise en place d'un système permettant de déterminer entre autres les relations de multi-coplanarité, de manière manuelle, semi-automatique ou automatique. Avant de donner nos résultats expérimentaux sur données réelles, nous présentons l'architecture du système complet de reconstruction que nous avons implanté.

Notons que deux séquences de scène plane par morceaux sont traitées en annexe A, où elles servent de tests pour la méthode de détection de plans qui y est présentée. Ces séquences sont "game system II" et la séquence du modem. Elle sont respectivement constituées de 6 et de 2 images.

5.7.2.1 Système de modélisation de scènes planes par morceaux

Nous présentons le système globalement puis en détaillons certaines phases.

Processus global. Le processus global de reconstruction d'une scène plane par morceaux est illustré sur le schéma de la figure 5.15. A partir des images, des points sont détectés, mis en correspondance et des contraintes de coplanarité sont définies. Ces étapes peuvent être manuelles ou automatiques. Pour les exemples présentés plus loin, nous avons défini un système semi-automatique, basé sur des homographies 2D, permettant d'établir simultanément des correspondances multi-vues et les contraintes de multi-coplanarité. Certains cas sont traités de manière complètement automatisée avec la méthode de détection de plans présentée en annexe A. Une fois les correspondances de points et les contraintes de multi-coplanarité communiquées au système, le reste du processus de reconstruction ne nécessite plus d'intervention manuelle. Il est constitué d'une étape d'initialisation, où une reconstruction sous-optimale est calculée, puis d'une étape d'ajustement de faisceaux, minimisant l'erreur de reprojection. Nous ne supposons pas le calibrage des caméras connu a priori. La reconstruction est donc effectuée en espace projectif, suivie d'une étape de calibrage en ligne des caméras, puis d'une seconde étape d'ajustement de faisceaux, cette fois en espace métrique. L'ajustement de faisceaux en espace projectif est optionnel. Cependant, il contribue à la réussite de l'étape de calibrage en ligne et à la bonne convergence de l'ajustement de faisceaux final en espace métrique. Si le calibrage des caméras est connu a priori, les étapes d'ajustement de faisceaux projectif et de calibrage en ligne peuvent être ignorées. L'ajustement de faisceaux

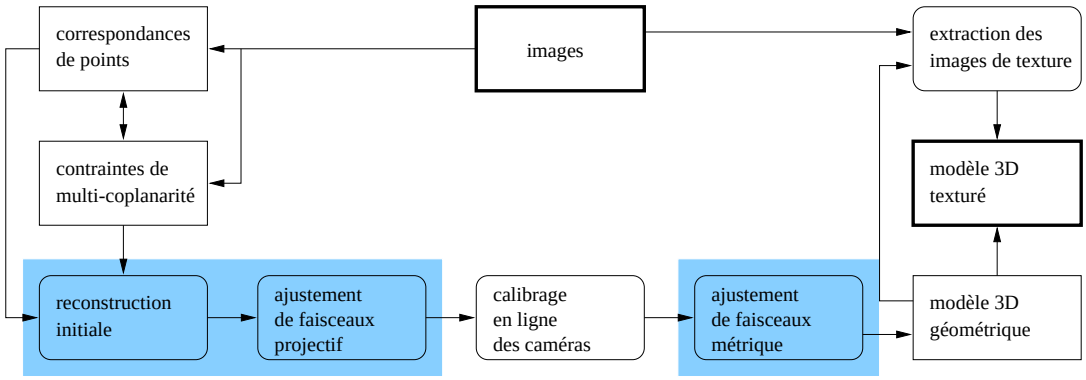


FIG. 5.15 – Schéma global de reconstruction d'une scène plane par morceaux à partir de caméras non calibrées. Les phases abordées en détail dans ce chapitre sont surlignées : la reconstruction initiale et de l'ajustement de faisceaux en espaces projectif et métrique.

métrique est suivi de l'extraction d'images de textures à partir des images originales.

Appariement de points et définition des contraintes. Les points sont détectés en utilisant le détecteur de Harris et Stephens [80] amélioré par Schmid et Mohr [172]. On supprime manuellement les points présentant peu d'intérêt pour la modélisation de la scène et on ajoute des points nécessaire à la délimitations des plans. L'appariement et les contraintes de multi-coplanarité sont entrés simultanément de manière semi-automatique à l'aide d'homographies 2D : l'utilisateur donne 4 correspondances de points, une homographie 2D est estimée et utilisée pour faire une mise en correspondance guidée telle qu'elle est décrite par Hartley et Zisserman [93, §3.8]. Dans certains des exemples montrés ci-dessous, d'autres processus d'appariement et de définition des contraintes de multi-coplanarité peuvent avoir été utilisés. Ils y sont décrit le cas échéant.

Processus de reconstruction. Nous suivons un processus de reconstruction basé sur les étapes suivantes. On commence par estimer une reconstruction de la scène en espace projectif, basé sur les méthodes d'initialisation plane par morceaux puis l'ajustement de faisceaux plan par morceaux décrit dans ce chapitre. Nous utilisons ensuite l'algorithme de calibrage en ligne de Pollefeys et al. [160] inspiré des travaux de Triggs [215]. Cet algorithme présente une singularité lorsque les axes optiques de toutes les caméras se coupent, ce qui est souvent le cas dans les applications de modélisation d'une scène. Nous modifions légèrement l'algorithme pour éviter cette singularité. Nous renvoyons à notre article [25] pour plus de détails sur cette modification. Un ajustement de faisceaux en espace métrique est ensuite lancé. La phase finale de reconstruction est une reconstruction

des points ne pouvant être reconstruits sans contrainte, par exemple, les points non observés et définis comme l'intersection de trois plans, où les points n'ajoutant pas de redondance d'information utile pour l'ajustement de faisceaux, tels les points observés dans une seule vue et contraints sur un plan. Nous renvoyons au tableau 5.1 pour un résumé des différents cas.

Extraction des images de texture. Afin d'obtenir un modèle 3D photoréaliste, nous plaquons des images texture sur les plans reconstruits. Ces images de texture sont extraites des images originales et corrigées perspectivement en utilisant les matrices de projection calibrées et une interpolation bicubique. Nous renvoyons par exemple aux travaux de Liebowitz et Zisserman [122] pour plus de détails et des algorithmes détaillés de correction perspective.

Evaluation de la qualité. Nous avons vu que l'erreur de reprojection ne reflètait pas la qualité de la reconstruction, et nous ne possédons pas la vérité terrain permettant de mesurer une erreur 3D comme dans le cas de données simulées. Nous effectuons certaines mesures métriques sur la reconstruction. Deux types de mesures sont significatives : les rapports de longueurs et les angles. Ces mesures sont faites sur les reconstructions non contraintes et celles obtenues par notre méthode plane par morceaux. Plus précisément, nous mesurons les variances σ_i des longueurs censées être égales et la moyenne des différences entre les angles mesurés et leur valeur réelle supposée, par exemple μ est la moyenne des $|1 - 2\alpha_i/\pi|$ pour les angles α_i censés être des angles droits.

5.7.2.2 Séquence de la maison

Nous présentons les résultats obtenus sur la séquence de la maison, dont les 6 images sont montrées en figure 5.16. Le tableau 5.5 montre l'erreur de reprojection à différents stades du processus de reconstruction. On

FIG. 5.16 – Les 6 images de la séquence de la maison.

remarque qu'en espace projectif, l'estimation contrainte converge vers une erreur de reprojection légèrement plus élevée que l'estimation non contrainte, ce qui est un résultat normal. Par contre, en espace métrique, l'erreur de reprojection donnée pour le modèle obtenu de manière contrainte est inférieure à celle obtenue de manière non contrainte. Cela signifie que la méthode non contrainte n'a pas convergé vers un minimum global et a convergé vers un minimum local assez mauvais. La figure 5.17 montre le modèle métrique obtenu avec la reconstruction contrainte et la figure 5.18 sa reprojection sur les images originales. Finalement, les images de texture sont extraites des images originales et le modèle photoréaliste montré en figure 5.19 est produit. Nous

espace	approche	étape	erreur de reprojection (pixels)	# itérations
projectif	non contrainte	initialisation	3,86	-
		ajustement de faisceaux	1,07	7
	contrainte	initialisation	1,90	-
		ajustement de faisceaux	1,20	3
métrique	non contrainte	ajustement de faisceaux	2,69	6
	contrainte	initialisation	3,86	-
		ajustement de faisceaux	1,43	9

TAB. 5.5 – Erreur de reprojection (pixels) et nombre d'itérations des ajustements de faisceaux à diverses étapes du processus de reconstruction pour la séquence de la maison.

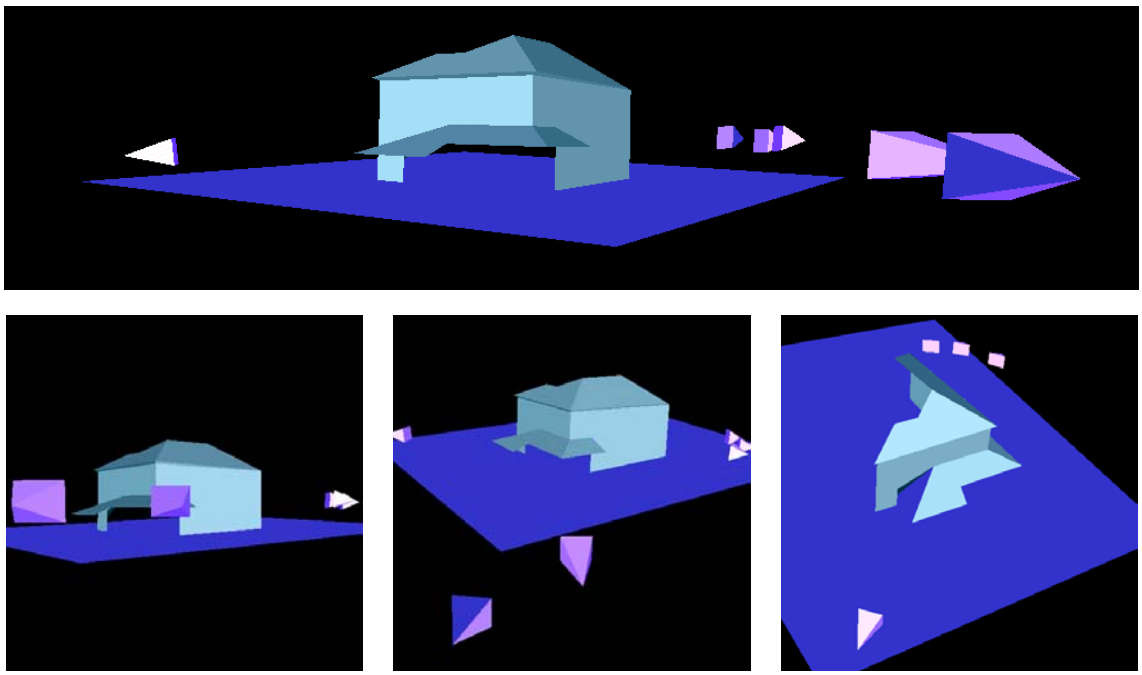


FIG. 5.17 – Structure et caméras métriques retrouvées par ajustement de faisceaux plan par morceaux. La structure est montrée comme un ensemble de polygones plans. Les caméras sont représentées par des pyramides. La hauteur d'une pyramide est proportionelle à la distance focale retrouvée pour la caméra. L'image en bas à droite montre un rendu d'un point de vue situé au dessus de la dernière image de la figure 5.16.

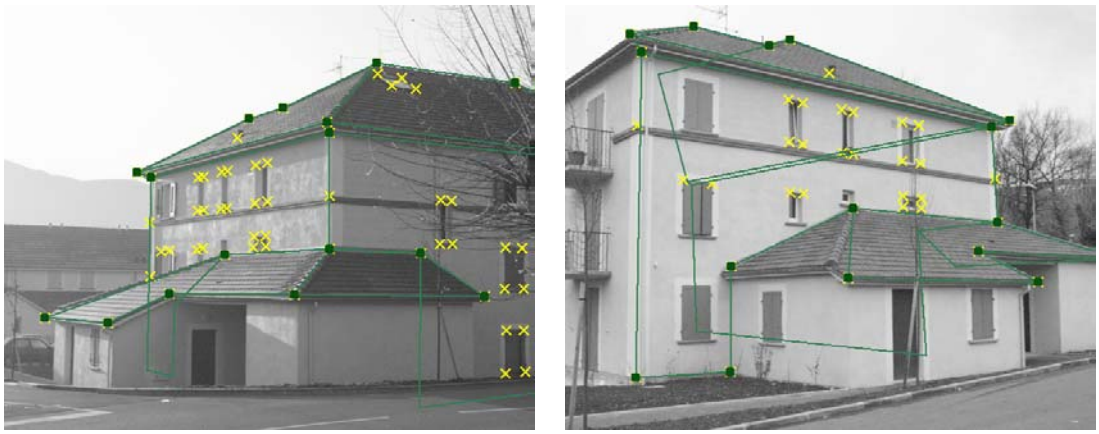


FIG. 5.18 – Reprojection du modèle métrique sur les images originales. Les croix indiquent les positions des points reprojetés.

FIG. 5.19 – Différentes vues du modèle métrique texturé. On note quelques artefacts introduits par des parties non planes non modélisées de la scène, par exemple le poteau sur le toît dans l'image en haut à droite est intégré au plan du toît et apparaît distordu dans les autres images, par exemple en haut à gauche.

	σ_1	σ_2	μ
non contraint	0,0489	0,0254	0,1032
contraint	0,0102	0,0168	0,0720

TAB. 5.6 – Mesures de qualité sur la reconstruction métrique à partir de la séquence de la maison, pour l'ajustement de faisceaux "non contraint" et "contraint". Plus les valeurs σ_1, σ_2 et μ (voir texte) sont faibles, meilleure est la reconstruction.

réalisons quelques mesures de qualité sur les modèles métriques obtenus par ajustement de faisceaux contraint et non contraint, reflétant les changements lorsque la méthode décrite dans ce chapitre est utilisée. Le tableau 5.6 montre ces mesures. On observe que la méthode contrainte produit une reconstruction avec des caractéristiques métriques meilleures que la reconstruction non contrainte.

5.7.2.3 Séquence "game system I"

Cette séquence est une paire stéréo constituée des deux images montrées sur la figure 5.1 (a) et (b). La modélisation plane par morceaux est montrée sur la figure 5.2. Nous utilisons les méthodes spécifiques dédiées au cas de deux vues, voir §5.4. Les erreurs de reprojection obtenues sont respectivement 1,24 et 0,95 pixels pour l'initialisation et l'ajustement de faisceaux plan par morceaux. Les figures 5.20 et 5.21 montrent respectivement

FIG. 5.20 – Deux vues du modèle projectif reconstruit pour la séquence "game system I".

les modèles projectif et métrique obtenus. Notons que l'ajustement de faisceaux non contraint converge avec une erreur de reprojection de 0,61 pixel, qui devient 2,32 pixels après ajustement de plans et correction des points. Nous effectuons des mesures de qualité métriques sur les reconstructions non contraintes de points et contrainte par multi-coplanarité. Ces mesures sont montrées dans le tableau sur la figure 5.22. Ces valeurs montrent que la qualité de la reconstruction obtenue avec la méthode contrainte est meilleure que celle de la reconstruction non contrainte.

5.7.2.4 Séquence des cubes

La séquence des cubes est une paire d'images d'une scène d'intérieur, montrées en figure 5.23. Nous définissons 82 correspondances de points, montrées sur la figure 5.24 (a), indiquant aussi le nombre de plans contenant chaque point. Nous définissons 9 plans pour modéliser la surface de la scène, montrés en figure 5.24 (b). Parmi ces 82 points, 2 ne sont sur aucun des plans, 68 sont sur un plan, 7 sur deux plans et 5 sur trois plans. Le nombre de paramètres de la structure est $9 \times 3 = 27$ pour les 9 plans, $2 \times 3 = 6$ pour les 2 points non contraints, $68 \times 2 = 136$ pour les 68 points sur un plan, $7 \times 1 = 7$ pour les 7 points sur deux plans et $5 \times 0 = 0$

FIG. 5.21 – Quelques vues du modèle métrique reconstruit pour la séquence “game system I”.

	σ_1	σ_2	μ
non contraint	0,0146	0,0543	0,0633
contraint	0,0082	0,0267	0,0413

FIG. 5.22 – Mesures de qualité sur la reconstruction métrique à partir de la séquence “game system I”, pour l’ajustement de faisceaux “non contraint” et “contraint”. Plus les valeurs de σ_1, σ_2 et μ (voir texte) sont faibles, meilleure est la reconstruction. Ces valeurs sont liées à la variance des longueurs censées être égales, les segments rouges d’une part et bleus d’autre part et les angles censés être droits.

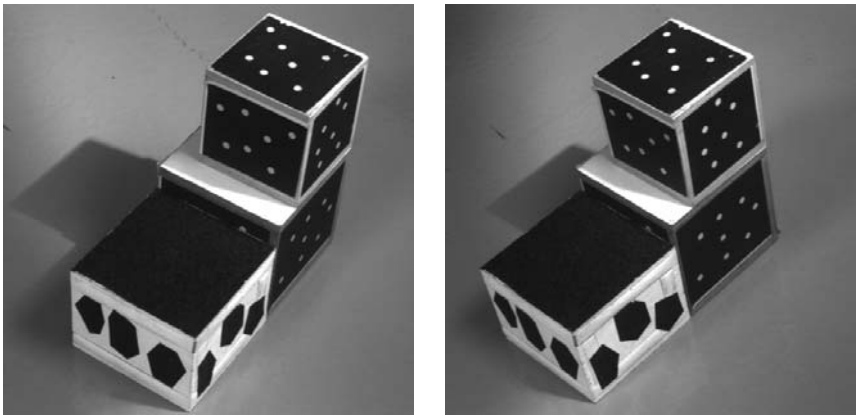
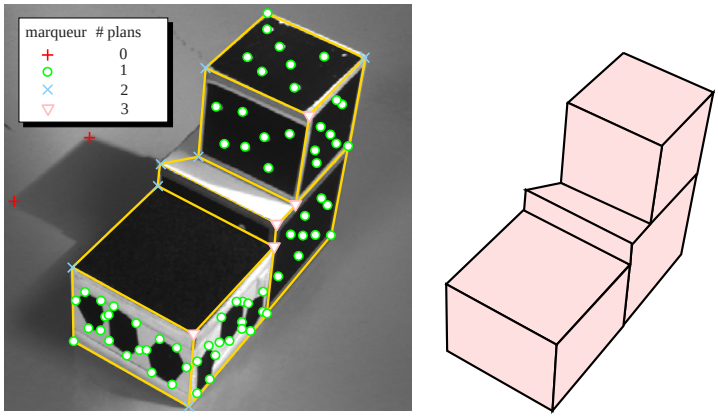


FIG. 5.23 – Séquence des cubes.



(a) (b)

FIG. 5.24 – (a) montre les points image utilisés avec le nombre de plans les contenant, ainsi que la délimitation des 9 plans considérés. (b) montre les plans modélisés.

pour les 5 points sur trois plans, soit un total de 176 degrés de liberté. Une modélisation non contrainte indépendante des 82 points a $82 \times 3 = 246$ paramètres. Notre paramétrisation contrainte minimale apporte donc une réduction de plus de 28% par rapport à une paramétrisation minimale individuelle de chaque point.

Nous appliquons notre méthode de reconstruction pour deux vues. L'erreur de reprojection obtenue pour la reconstruction contrainte est de 0,78 pixels puis de 0,56 pixels pour les phases d'initialisation et d'ajustement de faisceaux plan par morceaux respectivement. La figure 5.25 montre le modèle projectif obtenu. Notons que l'erreur de reprojection obtenue par ajustement de faisceaux non contraint est de 0,39 pixels. Cette erreur passe à 1,67 pixels, après ajustement de plans et correction des points sur ces plans. Nous utilisons la technique décrite

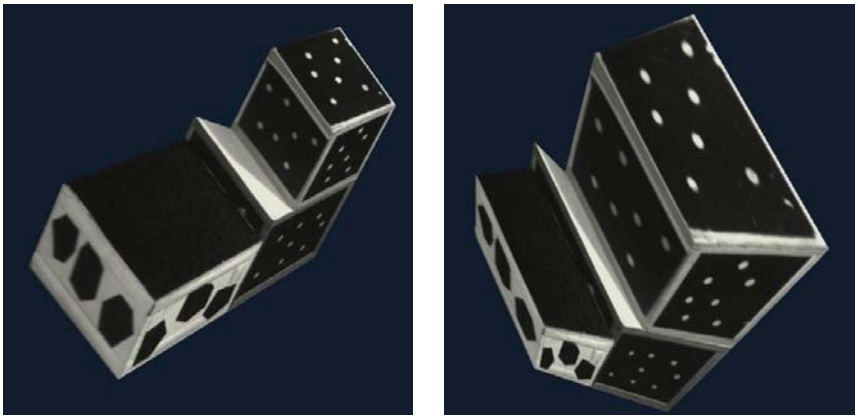


FIG. 5.25 – Modèle projectif obtenu pour la séquence des cubes.

dans [36] pour estimer les distances focales des caméras, avec l'hypothèse que la distortion entre les axes est zéro, le rapport d'aspect est un et le point principal est au centre de l'image. Ce calibrage permet de transférer le modèle projectif dans un espace métrique. La figure 5.26 montre le modèle métrique obtenu.

La figure 5.27 montre des mesures métriques sur les reconstructions non contraintes et contraintes. Dans ce tableau, σ_1 et σ_2 sont les variances des longueurs l_1, l_2 et l_3 et $l_4, \ldots, l_7$ respectivement. La valeur de μ est calculée comme la moyenne des $|1 - 2\alpha_i/\pi|$, où les α_i sont des angles censés être droits.

5.8 Conclusions et perspectives

Conclusions. Nous avons considéré le problème de l'utilisation de contraintes de multi-coplanarité pour la reconstruction. Ces contraintes permettent la modélisation de scènes planes par morceaux.

L'hypothèse que les points n'appartiennent pas à plus de trois plans permet de modéliser la structure comme un ensemble de plans indépendants et de points sur ces plans. Cette hypothèse simplifie la conception du système sans toutefois limiter les scènes modélisables : l'observation de points sur plus de trois plans est très rare. Elle permet la mise en œuvre de paramétrisations et d'algorithmes de reconstruction simples et performants.

Nous avons proposé des méthodes pour la reconstruction initiale. Contrairement à la méthode classique basée sur la reconstruction non contrainte des points, l'ajustement de plans puis la correction des points, nos méthodes reconstruisent les plans puis des points sous contraintes de multi-coplanarité. Elles sont linéaires et minimisent des critères d'erreur basés image. Les points sont reconstruits par des méthodes de *triangulation contrainte par multi-coplanarité*.

Nous avons ensuite étudié l'incorporation des contraintes de multi-coplanarité à un système d'ajustement de faisceaux. Le problème original, formulé comme la minimisation de l'erreur de reprojection sous des contraintes non-linéaires, est transformé en un problème d'ajustement de faisceaux standard, en utilisant une *paramétrisation minimale* de la structure incorporant les contraintes géométriques. Cet *ajustement de faisceaux plan par morceaux* est optimal car il minimise l'erreur de reprojection en garantissant les contraintes géométriques.

Nous avons utilisé des données réelles et simulées pour comparer nos algorithmes aux méthodes existantes. La comparaison avec la vérité terrain et la mesure de quantités métriques révèlent que l'utilisation des

contraintes de multi-coplanarité améliore la qualité de la reconstruction par rapport à un ajustement de faisceaux non contraint. Ceci montre aussi que l'utilisation d'une méthode optimale comme celle que nous proposons améliore la précision de la reconstruction comparé à d'autres méthodes (sous-optimales). Nous avons évalué expérimentalement les limites de validité des contraintes de multi-coplanarité lorsque la planarité par morceaux est perturbée. Cette étude révèle que dans une grande majorité des conditions d'observation, ces contraintes permettent d'améliorer la qualité de la reconstruction.

Perspectives. Une piste d'extension des méthodes proposées concerne l'utilisation de droites, en préservant la simplicité et l'optimalité de notre approche. La simplicité est basée sur l'hypothèse qu'un point n'est pas sur plus de trois plans. Les contraintes peuvent donc être vues comme exercées par les plans sur les points. L'incorporation de droites pourrait être effectuée en imaginant un schéma où les plans contraindraient les droites et les points ; et les droites contraindraient les points, basé par exemple sur les hypothèses que deux droites concourantes devraient appartenir au même plan, et qu'un point ne pourrait pas être sur plus de deux droites (mis à part les bords des plans). Les droites pourraient alors être modélisées indépendamment des points. On pourrait modéliser les plans comme précédemment, certains points sur les plans, puis des droites sur des plans, et enfin des points sur des droites. Un tel ordre de dépendance devrait permettre de concevoir des méthodes de reconstruction initiale efficaces, et des paramétrisations simples pour l'ajustement de faisceaux.

Finalement, une perspective de travail est l'étude d'un système de reconstruction de scènes planes par morceaux, à partir d'une séquence d'images continue. L'intérêt d'un tel système serait d'obtenir une reconstruction très précise, car basée sur beaucoup d'images, et de manière automatique, par suivi des primitives ou des plans au travers de la séquence. On pourrait utiliser un système de suivi de plan, voir par exemple le travail récent de Jurie et Dhome [105], couplé à une estimation consistante de la géométrie, c'est à dire aux contraintes multi-vues entre les homographies de plan.

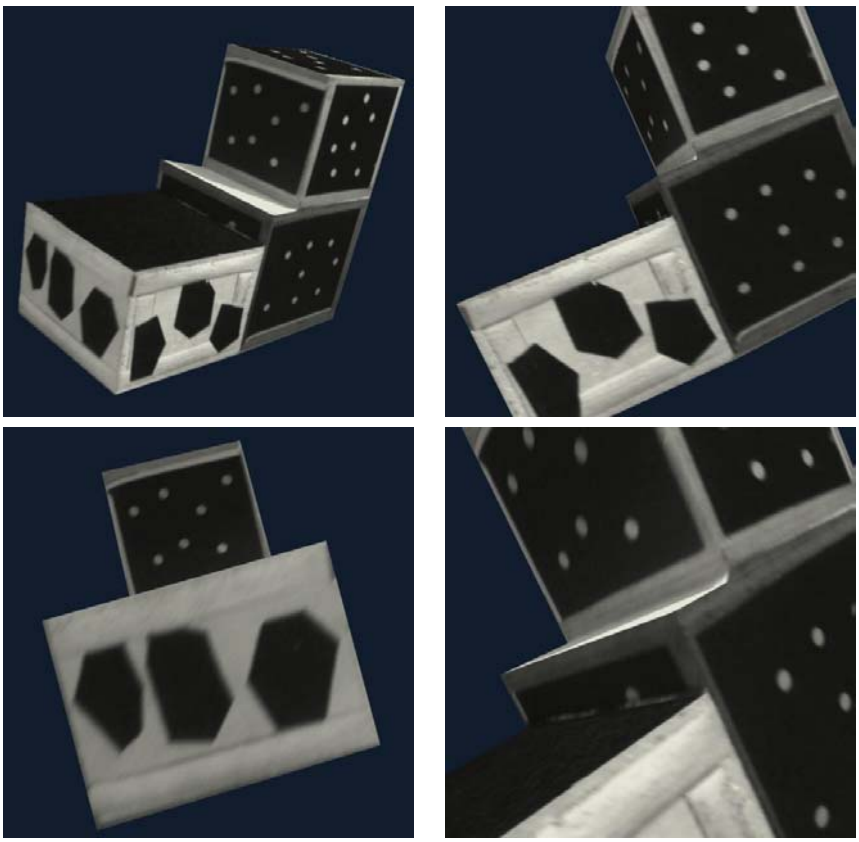
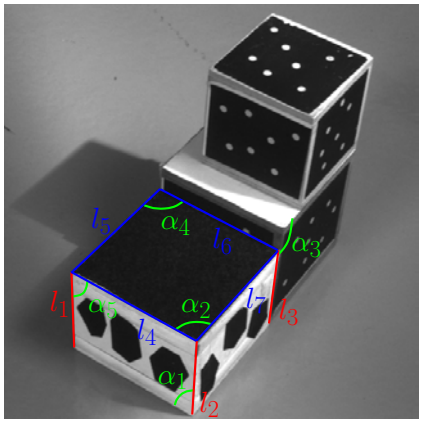


FIG. 5.26 – Quelques vues du modèle métrique obtenu pour la séquence des cubes.



	σ_1	σ_2	μ
non contraint	0,0039	0,0018	0,0721
contraint	0,0020	0,0000	0,0134

FIG. 5.27 – Mesures de qualité sur les reconstructions métriques obtenues par ajustement de faisceaux “non contraint” ou “contraint” par multi-coplanarité. Plus les valeurs de σ_1, σ_2 et μ (voir texte) sont faibles, meilleure est la reconstruction. Ces valeurs sont liées à la variances des longueurs censées être égales, les segments rouges d’une part et bleus d’autre part et les angles censés être droits.

CHAPITRE

6

RECONSTRUCTION DE DROITES ET DE CAMÉRAS

Ce chapitre est consacré à la reconstruction de droites et de caméras à partir de correspondances multi-vues. Nous abordons les problèmes de la reconstruction initiale et de l'ajustement de faisceaux final. La difficulté principale est la représentation algébrique des droites 3D : il n'existe pas de représentation "naturelle", ni de paramétrisation non singulière, minimale et simple de leurs 4 degrés de liberté. Le chapitre commence par une étude de différentes représentations, évaluées en vue de leur utilisation dans des algorithmes linéaires et non-linéaires.

Concernant la reconstruction initiale, nous adoptons une approche basée sur la reconstruction des caméras puis des droites par triangulation. Nos contributions concernent la triangulation. L'étude précédente des représentations oriente notre choix sur les coordonnées de Plücker. Une propriété clef est que les coordonnées des droites reprojetées sont linéaires en termes des coordonnées de Plücker. Nous proposons un algorithme de triangulation itératif simple minimisant l'erreur de reprojection.

Concernant l'ajustement de faisceaux, les représentations existantes induisent des paramétrisations inadaptées, dans le sens où elles sont partielles (les droites 3D ne sont pas toutes représentées) ou soumises à des contraintes de consistance interne, par exemple la contrainte de Plücker. Nous proposons la *représentation orthonormale des droites 3D*. Elle permet d'effectuer l'optimisation non-linéaire sur un nombre minimal de 4 paramètres, sans contrainte interne ni liberté de jauge.

Finalement, nos algorithmes sont validés sur des données simulées et réelles.

Ces travaux ont été effectués en collaboration avec Peter Sturm et ont été publiés dans [26].

6.1 Introduction

Le but de ce chapitre est la conception de méthodes pour la reconstruction de droites à partir de correspondances entre plusieurs vues. Notre étude et nos propositions portent sur la reconstruction initiale et l'ajustement de faisceaux final. Les droites sont utilisées dans de nombreux domaines tels la visualisation scientifique ou la modélisation géométrique. Leur étude fait l'objet du livre de Pottmann et Wallner [161]. Si la primitive la plus utilisée en vision par ordinateur et en photogrammétrie est le point, la plupart des problèmes étudiés avec des points le sont aussi avec des droites lorsque cela est possible. On trouve entre autres des solutions pour la détection dans une image [39], le suivi dans une séquence d'image [77, 234] et l'appariement [173]. On trouve des applications telles le calcul de pose [125], l'asservissement visuel [4, 151] et l'estimation des distorsions optiques d'une caméra [162].

Les avantages à utiliser des droites sont multiples. C'est une primitive souvent présente, notamment en environnements humains. Sa localisation dans les images, et particulièrement la détermination de la direction, est plus précise que celle des points. Sa mise en correspondance est plus fiable que celle des points et elle est plus utile pour des tâches du "haut niveau" telle la reconnaissance d'objets. Les inconvénients sont que contrairement aux points, il n'existe pas de représentation algébrique naturelle des droites 3D, et que la reconstruction de droites et de caméras à partir de deux vues n'est pas possible. En outre, l'erreur de reprojection, souvent minimisée par les algorithmes de reconstruction, est moins naturelle à formuler sur les droites que sur les points, car il n'existe pas de distance Euclidienne pour ces dernières.

En vision par ordinateur, la reconstruction de caméras et de droites a une histoire liée à la découverte du tenseur trifocal, voir en particulier les travaux de Hartley [92] et de Shashua [178]. Il existe de nombreux travaux portant sur la reconstruction de trois caméras à partir de correspondances de droites, entre autres ceux de Liu et Huang [123, 124], Spetsakis et al. [187, 188], Weng et al. [226] et Navab et al. [153]. La géométrie multi-vues des droites à été bien étudiée, voir par exemple les travaux de Faugeras et al. [62] et de Hartley et al. [93]. Les différentes méthodes de reconstruction multi-vues sont souvent inspirées des algorithmes basés sur des points, et héritent de caractéristiques similaires. La section 3.2 du chapitre 3 passe en revue les différentes méthodes basées sur des points, qui apparaissent comme séquentielles ou directs. Les algorithmes directs traitent toutes les vues simultanément alors que les algorithmes séquentiels procèdent par sous-ensembles de vues. Il existe de nombreux algorithmes séquentiels pour la reconstruction de caméras et de droites [142, 155, 175, 203, 225], en particulier les algorithmes de Ayache et Faugeras [56] et de Navab et Zhang [154].

Les algorithmes directs sont basés sur la factorisation ou les contraintes de fermeture. On renvoie à la section 3.2.1 du chapitre 3 pour plus de détails sur ces méthodes. La méthode de factorisation étendue de la version pour les points en une version pour les droites par Quan et Kanade [164] (modèle de caméra affine) et par Triggs [214] puis Martinec et Pajdla [135] (modèle de caméra perspectif) nécessite que les droites considérées soient visibles dans toutes les vues. La méthode basée sur les contraintes de fermeture proposée par Triggs [216] pour un modèle de caméra perspectif puis spécialisée au modèle de caméra affine par Heyden et Kahl [99] permet d'obtenir toutes les caméras (non calibrées) en résolvant un système linéaire construit à partir des tenseurs d'appariement multi-vues. La structure est ensuite reconstruite par triangulation. Notons que Govindu [72] propose une approche similaire à celle de Triggs mais pour des caméras calibrées.

En photogrammétrie, Mulawa et Mikhail [150] introduisent le concept de l'utilisation de droites comme observations à la place des points. Depuis, de nombreux travaux basés sur les droites ont été publiés dans ce domaine, voir entre autres Habib et al. [75, 76], Tommaselli et al. [205], Heikkinen [94], Kubik [114] et Roberts [166]. Les différences entre ces travaux portent principalement sur le choix de la représentation algébrique des droites 3D et de la fonction de coût minimisée.

Ce chapitre a des implications en vision par ordinateur et en photogrammétrie. On y présente une méthode de *triangulation* multi-vues (le problème de l'*intersection* en photogrammétrie) auto-initialisante permettant la reconstruction initiale des droites ainsi qu'une nouvelle représentation des droites 3D, la *représentation orthonormale*, dont la caractéristique principale est de permettre une reconstruction par ajustement de faisceaux impliquant le minimum de 4 paramètres par droite reconstruite. Ces méthodes minimisent toutes deux l'erreur de reprojection, garantissant sous certaines hypothèses une solution optimale. Nous ne faisons pas d'hypothèse sur le calibrage des caméras. Nous supposons que les correspondances de droites considérées donnent assez de

contraintes pour permettre la reconstruction.

Nous ne traitons pas la reconstruction initiale des caméras. Ces dernières peuvent être reconstruites en utilisant par exemple les contraintes de fermeture de Triggs [216] induites par les tenseurs trifocaux entre triplets d'images, dont l'estimation est possible à partir de correspondances de droites, par exemple avec l'algorithme de Hartley [92]. Un schéma d'estimation robuste permettant d'estimer le tenseur trifocal tout en réalisant la mise en correspondance des droites est donné par Torr et Zisserman [211]. L'appariement de droites a été étudié en particulier dans [173]. Dans le domaine de la photogrammétrie, les positions des caméras sont souvent déterminées par GPS ou par une centrale inertielle. Nous n'abordons pas non plus le problème des configurations singulières pour la reconstruction de caméras à partir de correspondances de droites, pour lequel nous renvoyons aux travaux de Buchanan [38] et de Navab et Faugeras [152].

La reconstruction de droites par triangulation et ajustement de faisceaux est abordée en détails dans ce chapitre. Ces problèmes soulèvent la difficulté principale de l'utilisation de droites : leur représentation algébrique. En effet, il n'existe pas de représentation minimale, complète et globalement non singulière de l'espace de dimension 4 des droites 3D, voir par exemple [93, §2.2]. En conséquence, les droites 3D sont la plupart du temps surparamétrées. Différentes représentations et leurs caractéristiques sont examinées en §6.2. Bien sûr, la représentation la mieux adaptée dépend du problème considéré. Par exemple, l'algorithme de calcul optimal du tenseur trifocal donné par Hartley [92] est basé sur une représentation par deux droites image, alors que l'algorithme de reconstruction séquentiel de Seo et al. [175] est basé sur les coordonnées de Plücker.

La triangulation, consiste à reconstruire les droites étant donnée une reconstruction des caméras. La triangulation de chaque droite est indépendante. Il existe peu de travaux sur la triangulation multi-vues de droites car ces dernières sont souvent reconstruites simultanément aux caméras. Hartley [92] propose une méthode de triangulation linéaire basée sur une représentation des droites 3D par deux points 3D. Son algorithme minimise une erreur algébrique entre les points reprojetés et les droites image observées.

L'ajustement de faisceaux, est une procédure itérative impliquant les paramètres des caméras et des droites, et minimisant l'erreur de reprojection. Le chapitre 3 décrit des méthodes numériques standard d'ajustement de faisceaux. Les représentations existantes des droites 3D ne sont pas bien adaptées aux algorithmes d'optimisation non-linéaire. Par exemple, les paramétrisations par deux points ou deux plans ont 4 degrés de liberté de jauge chacune[1], ce qui peut entraîner des instabilités numériques. La paramétrisation basée sur deux droites image a 2 degrés de liberté de jauge, implique le choix d'images où la droite considérée est visible et admet d'importantes singularités. L'estimation directe de coordonnées de Plücker n'a du sens que si la technique d'optimisation garantit la contrainte de Plücker. Une représentation appropriée n'implique ni contraintes internes, ni libertés de jauge.

Contributions. Nos contributions portent sur la triangulation et l'ajustement de faisceaux. Concernant la triangulation, nous montrons qu'avec une représentation des droites 3D en coordonnées de Plücker, on obtient une solution linéaire minimisant une erreur algébrique, dont la forme est proche de l'erreur de reprojection. Ceci n'est possible que si la contrainte de Plücker est ignorée puis appliquée à l'issue de l'estimation sur le vecteur calculé, de manière similaire à l'estimation de la matrice fondamentale par l'algorithme des huit points où la contrainte de singularité est ignorée puis appliquée à l'issue de l'estimation. Cette approche présente deux biais. Le premier biais est introduit par l'utilisation d'une erreur algébrique, par rapport à l'erreur de reprojection. Le deuxième biais est dû à la non prise en compte directe de la contrainte de Plücker. Nous proposons des solutions pour corriger ces deux biais. Notre algorithme est simple, minimise l'erreur de reprojection et ne nécessite pas d'initialisation externe. Le premier biais est corrigé par un schéma d'optimisation de type quasi-linéaire. Le deuxième biais est corrigé par l'incorporation au premier ordre de la contrainte de Plücker dans l'estimation.

Concernant l'ajustement de faisceaux, nous proposons une *représentation orthonormale* des droites 3D. Cette représentation est complète, c'est à dire que toutes les droites 3D sont représentées. Elle permet d'effectuer l'ajustement de faisceaux avec le nombre minimal de 4 paramètres pour chaque droite 3D. Ces derniers ne sont donc pas soumis à des contraintes internes et n'ont pas de liberté de jauge. Ils correspondent géométriquement

[1] Pour les deux points : leur position le long de la droite (jauge géométrique) et le facteur d'échelle pour leur représentation en coordonnées homogènes (jauge algébrique).

d'une part, à l'orientation d'une base orthonormale donnant entre autres la direction de la droite et d'autre part, à la distance entre la droite et l'origine.

Organisation du chapitre. Nous passons en revue plusieurs représentations des droites 3D en §6.2. Nous abordons ensuite le problème de la triangulation en §6.3 puis celui de l'ajustement de faisceaux en §6.4. Nous donnons des résultats expérimentaux en §6.5 puis concluons et donnons des perspectives de travail en §6.6.

6.2 Représentation des droites 3D

Nous décrivons différentes représentations des droites 3D et leurs caractéristiques. Certaines de ces représentations sont partielles, c'est à dire qu'elles ne représentent qu'un sous-ensemble des droites 3D. Par exemple, certains travaux sur la reconstruction métrique, et particulièrement en photogrammétrie, supposent que les droites reconstruites ne sont pas à l'infini. Le but de cette étude est de déterminer une représentation pour la triangulation et pour l'ajustement de faisceaux. Concernant la triangulation, le critère déterminant est que la reprojection de la droite soit linéaire. L'ajustement de faisceaux est une procédure d'optimisation non-linéaire itérative qui permet de disposer de plus de souplesse dans le choix de la paramétrisation. La qualité de la paramétrisation induite par une représentation est estimée selon des critères tels le nombre de degrés de liberté de jauge ou de contraintes internes, discutés dans la section 3.5.4 du chapitre 3. Les résultats de cette étude sont résumés dans le tableau 6.1. Nous décrivons des représentations complètes, permettant de représenter toutes les droites, et des représentations partielles, présentant des singularités. La première représentation décrite est les coordonnées de Plücker. Le lien entre chaque autre représentation et les coordonnées de Plücker est ensuite établi.

6.2.1 Représentations complètes

Coordonnées de Plücker. Les coordonnées de Plücker [159] sont décrites en détail au chapitre 2, §2.12. Brièvement, elles sont constituées par en un vecteur-6 de coordonnées homogènes noté $\mathbf{L}^{\mathsf{T}} \sim (\ \mathbf{a}^{\mathsf{T}} \quad \mathbf{b}^{\mathsf{T}}\ )$ soumis à la contrainte bilinéaire $\mathbf{a}^{\mathsf{T}}\mathbf{b} = 0$, appelée la contrainte de Plücker. Les coordonnées de Plücker ont donc un degré de liberté de jauge algébrique (le facteur d'échelle) et une contrainte de consistance interne (la contrainte de Plücker). Elles s'interprètent géométriquement comme suit. La partie supérieure, le vecteur-3 $\mathbf{a}$, représente la direction de la normale au plan formé par la droite et l'origine. La partie inférieure, le vecteur-3 $\mathbf{b}$, est le point à l'infini de la droite, c'est à dire sa direction. L'équation de la droite reprojetée est linéaire en les coordonnées de Plücker. La matrice (3×6) de projection pour coordonnées de Plücker est formulée en §6.3.3. Seo et al. [175] utilisent cette représentation dans un algorithme de reconstruction séquentiel.

Deux points ou deux plans. Ces deux représentations sont duales. De nombreux détails sont donnés dans [93, §2.2.2]. La première représentation consiste à utiliser deux points $\boldsymbol{M}$ et $\boldsymbol{N}$ sur la droite, et la deuxième représentation consiste à définir la droite comme l'intersection de deux plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi'}$. Ces représentations ont des caractéristiques similaires. Elles ont 8 paramètres, donc 4 degrés de liberté de jauge, la position des points le long de la droite (respectivement la position des plans dans le faisceaux de plans autour de la droite) et les facteurs d'échelle dans la représentation en coordonnées homogènes des points ou plans. Dans le cadre de la reconstruction métrique, si l'on exclut les droites à l'infini, la représentation avec deux points peut être formulée avec 6 paramètres. Le lien avec les coordonnées de Plücker est direct, par les définitions (2.54) et (2.56) en fonction des points et des plans respectivement. La reprojection $\boldsymbol{l}$ de la droite par une caméra de matrice P est une fonction bilinéaire des paramètres. Par exemple, pour la représentation avec deux points, $\mathbf{l} \sim (\mathrm{P}\mathbf{M}) \wedge (\mathrm{P}\mathbf{N})$. Hartley [92] propose un algorithme linéaire de triangulation basé sur cette représentation. Habib et al. [76] utilisent cette représentation pour l'ajustement de faisceaux. Ils considèrent que la droite est finie. L'ambiguïté sur la position des deux points le long de la droite est fixée en considérant qu'ils correspondent aux extrémités observées dans une des images.

6.2.2 Représentations partielles

Point le plus proche et direction. Les algorithmes de reconstruction de droites et de caméras pour 3 vues de Weng et al. [226] et pour vues multiples de Taylor et Kriegman [203] utilisent la représentation suivante. Chaque droite est représentée par son point le plus proche de l'origine de coordonnées $\mathbf{Q}^\mathsf{T} \sim (\ \bar{\mathbf{Q}}^\mathsf{T} \quad 1)$ et sa direction de coordonnées $\mathbf{Q}_\infty^\mathsf{T} \sim (\ \bar{\mathbf{Q}}_\infty^\mathsf{T} \quad 0)$, ce qui donne 6 paramètres. Cette représentation exclut les droites à l'infini et n'est utilisable qu'en espace Euclidien ou métrique[2]. Elle est reliée aux coordonnées de Plücker $\mathbf{L}$ de la droite par :

$$\mathbf{L} \sim \begin{pmatrix} \bar{\mathbf{Q}} \wedge \bar{\mathbf{Q}}_\infty \\ \bar{\mathbf{Q}}_\infty \end{pmatrix}.$$

La reprojection $\boldsymbol{l}$ de la droite par une caméra de matrice $\mathsf{P} \sim (\ \bar{\mathsf{P}} \quad \mathbf{p})$ est une fonction bilinéaire des paramètres : $\mathbf{l} \sim (\bar{\mathsf{P}}\bar{\mathbf{Q}} + \mathbf{p}) \wedge (\bar{\mathsf{P}}\bar{\mathbf{Q}}_\infty)$. En photogrammétrie, Tommaselli et Lugnani [205] utilisent cette représentation pour l'ajustement de faisceaux. Mulawa et Mikhail [150] s'en servent avec la contrainte $\|\bar{\mathbf{Q}}_\infty\| = 1$.

Deux Projections. Il est possible de représenter une droite 3D par deux projections [92, 187]. Ceci est lié au fait que la reconstruction d'une droite à partir de deux images admet une solution unique.

Spetsakis et Aloimonos [187] utilisent l'intersection de deux plans, l'un parallèle au plan $x = 0$ et l'autre parallèle au plan $y = 0$, ce qui est équivalent aux projections de la droite sur les plans $x = 0$ et $y = 0$. Ces deux plans sont formulés en fonction de 4 paramètres a, b, c et d par :

$$\begin{cases} x &= az + b \\ y &= cz + d. \end{cases}$$

L'espace de dimension 1 des points $\boldsymbol{Q}$ sur la droite est paramétré par la coordonnée z :

$$\mathbf{Q} \sim \begin{pmatrix} az + b \\ cz + d \\ z \\ 1 \end{pmatrix}.$$

Cette représentation admet des singularités évidentes : les droites parallèles au plan $z = 0$ ne peuvent pas être représentées. En effet, les points d'une telle droite ont un z constant. Les points étant paramétrés par z, on obtient toujours le même point si z est constant. On peut relier cette représentation aux coordonnées de Plücker $\mathbf{L}$ de la droite en considérant deux points quelconques sur la droite, par exemple pour $z = 0$ et $z = 1$, puis la définition (2.54) des coordonnées de Plücker, ce qui donne :

$$\mathbf{L} \sim \begin{pmatrix} d \\ -b \\ bc - da \\ a \\ c \\ 1 \end{pmatrix}.$$

Ayache et Faugeras [56] utilisent de même cette représentation. En photogrammétrie, Habib [75] étend cette représentation en utilisant des paires de plans différentes selon la droite, afin de réduire les singularités.

Hartley [92] utilise deux images de la droite. Cette représentation admet les singularités suivantes : toutes les droites contenues dans un plan épipolaire induit par les deux caméras ont les mêmes images dans ces deux vues. Les droites 3D non représentables forment donc un complexe linéaire ("Linear Line Complex" en Anglais,

[2] La notion de point le plus proche n'existe pas en espaces affine et projectif.

voir par exemple [189]) dont la base est formée par la droite entre les deux centres de projection. Notons que ces singularités peuvent être facilement rencontrées en pratique. Les coordonnées de Plücker correspondants à cette représentation peuvent être calculées en intersectant les deux plans de vue induits par les deux droites image, selon l'équation (2.56). Hartley montre que la reprojection de la droite dans d'autres vues s'exprime de manière bilinéaire.

6.2.3 Résumé

Le tableau 6.1 résume les caractéristiques des différentes représentations étudiées ci-dessus, et de la représentation orthonormale que nous proposons en §6.4.2. On constate que la seule représentation permettant

représentation	complète	ddl de jauge	contraintes	reprojection	maj minimale
point le plus proche et direction	non	1	1	bilinéaire	non
deux droites image	non	2	0	bilinéaire	non
deux points ou deux plans	oui	4	0	bilinéaire	non
coordonnées de Plücker	oui	1	1	linéaire	non
représentation orthonormale	oui	0	0	bilinéaire	oui

TAB. 6.1 – Résumé de différentes représentations de droites 3D et de leurs caractéristiques. Notons que le nombre de degrés de liberté de jauge et de contraintes internes sont étroitement liés. La colonne "reprojection" indique la forme de l'équation de la droite reprojetée en termes des paramètres de la représentation. La colonne "maj (mise à jour) minimale" indique si la représentation est mise à jour avec le nombre minimal de 4 paramètres. La représentation orthonormale des droites 3D est décrite en §6.4.2.

d'obtenir l'équation de reprojection comme une fonction linéaire des paramètres est les coordonnées de Plücker. Un autre constat est qu'à part la représentation orthonormale, aucune représentation ne permet une mise à jour minimale, ce qui est dû à des libertés de jauge et / ou une contrainte de consistance interne. Une mise à jour minimale est un critère important pour l'utilisation d'une représentation dans un algorithme d'ajustement de faisceaux.

6.3 Triangulation

Nous discutons le problème du calcul de la structure étant donné le mouvement des caméras. Nous rappelons l'expression de l'erreur de reprojection donnée au chapitre 3, §3.3.2. Pour une droite $\boldsymbol{L}$ et un ensemble de caméras $\mathcal{M} = \{\mathsf{P}_1, \ldots, \mathsf{P}_n\}$, elle s'exprime par :

$$\mathcal{L}(\boldsymbol{L}, \mathcal{M}) \quad = \quad \sum_{i=1}^{n} \left(d^2_{PH}(\mathbf{x}_i, \hat{\mathbf{l}}_i) + d^2_{PH}(\mathbf{y}_i, \hat{\mathbf{l}}_i) \right). \tag{6.1}$$

où d^2_{PH} est la distance Euclidienne point-droite et $\mathbf{x}_i$ et $\mathbf{y}_i$ sont deux points sur la $i^{\text{ème}}$ image de la droite. Les droites reprojetées sont notées $\hat{\mathbf{l}}_i$. Leur expression dépend de la représentation choisie pour la droite 3D. Notons que lorsque la droite image est définie comme le meilleur ajustement à un ensemble de points, on peut toujours exprimer l'erreur de reprojection en termes de la somme de deux distances Euclidienne pondérées (la pondération n'est pas montrée explicitement dans l'équation (6.1)). Pour simplifier les notations, et sans perte de généralité, nous supposons que $\boldsymbol{L}$ est visible dans les n vues.

Nous présentons et discutons l'algorithme linéaire de Hartley [92] puis étudions une méthode linéaire et des méthodes itératives pour la minimisation de l'erreur de reprojection. Ces algorithmes sont généraux dans le sens où ils peuvent être utilisés avec des caméras calibrées ou non, c'est à dire que des reconstructions projective, affine, métrique ou Euclidienne peuvent être obtenues.

6.3.1 Travaux précédents

La plupart des travaux existants sur la triangulation des droites considèrent le cas de trois vues et proposent des solutions analytiques difficilement généralisables au cas multi-vues, voir par exemple [226]. Nous présentons et discutons l'algorithme linéaire de Hartley [92]. Cette approche est basée sur les représentations duales de la droite (c'est à dire par deux points ou par deux plans) et s'applique au cas multi-vues. Nous la désignons par LIN_HARTLEY.

Intersection de plans. Lorsque deux vues sont utilisées, l'intersection des deux plans de vue associés aux droites observées donne une solution exacte pour la droite 3D, quelle que soit la représentation choisie. Lorsque plus de deux vues sont disponibles, Hartley [92] propose d'utiliser la représentation par deux points avec le calcul suivant. Notons $\boldsymbol{l}_i$ les droites observées dans les images. Les plans de vue $\boldsymbol{\Pi}_i$ correspondants sont de coordonnées $\mathbf{\Pi}_i \sim \mathsf{P}_i^\mathsf{T}\mathbf{l}_i$. En l'absence de bruit, les coordonnées de tout point $\boldsymbol{Q}$ sur la droite satisfont les équations $\mathbf{\Pi}_i^\mathsf{T}\mathbf{Q} \sim \mathbf{l}_i^\mathsf{T}\mathsf{P}_i\mathbf{Q} = 0$, induisant le système linéaire suivant :

$$\mathsf{A}_{(n\times 4)}\mathbf{Q}_{(4\times 1)} \;=\; \mathbf{0}_{(n\times 1)} \quad \text{avec} \quad \mathsf{A} \;=\; \begin{pmatrix} \cdots \\ \mathbf{l}_i^\mathsf{T}\mathsf{P}_i \\ \cdots \end{pmatrix}. \tag{6.2}$$

En l'absence de bruit, le rang de la matrice A est deux, mais en présence de bruit il est plein. Les deux vecteurs singuliers associés aux deux valeurs singulières minimales de A s'interprètent comme les coordonnées de deux points $\boldsymbol{M}$ et $\boldsymbol{N}$ représentant la droite reconstruite.

Une variante de cet algorithme consiste à retrouver deux plans représentant la droite comme les deux vecteur singuliers de A associés à ses deux valeurs singulières maximales, voir le livre de Hartley et Zisserman [93, §11.6]. Notons que ces deux solutions sont strictement équivalentes, pour la raison suivante. Soit $\mathsf{A}_{(n\times 4)} \sim \mathsf{U}_{(n\times 4)}\mathsf{\Sigma}_{(4\times 4)}\mathsf{V}_{(4\times 4)}^\mathsf{T}$ la décomposition en valeurs singulières de la matrice A. Les colonnes de la matrice V sont les vecteurs singuliers de la matrice A, ordonnés tels que les valeurs singulières correspondantes soient en ordre décroissant. La solution originale de Hartley consiste à interpréter les deux dernières colonnes de la matrice V comme les coordonnées homogènes de deux points $\boldsymbol{M}$ et $\boldsymbol{N}$ représentant la droite. La variante de l'algorithme donnée dans le livre de Hartley et Zisserman consiste à interpréter les deux premières colonnes de cette matrice comme les coordonnées homogènes de deux plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi'}$ contenant la droite. On peut résumer ceci par :

$$\mathsf{V} \;\sim\; (\mathbf{\Pi}\ \mathbf{\Pi}'\ \mathbf{M}\ \mathbf{N}).$$

La matrice V étant orthonormale, nous avons :

$$\mathbf{\Pi}^\mathsf{T}\mathbf{M} \;=\; \mathbf{\Pi}^\mathsf{T}\mathbf{N} \;=\; \mathbf{\Pi}'^\mathsf{T}\mathbf{M} \;=\; \mathbf{\Pi}'^\mathsf{T}\mathbf{N} \;=\; 0.$$

Ceci s'interprète géométriquement comme le fait que les plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi'}$ contiennent les points $\boldsymbol{M}$ et $\boldsymbol{N}$. Ils définissent donc la même droite.

Analyse de l'erreur. Lors de la résolution du système (6.2) au sens des moindres carrés, les coordonnées du point $\boldsymbol{M}$, données par le vecteur singulier associé à la plus petite valeur singulière de A, satisfont l'équation suivante :

$$\min_{\mathbf{M},\|\mathbf{M}\|^2=1} E(\mathbf{M},\mathcal{M}) \quad \text{avec} \quad E(\mathbf{Q},\mathcal{M}) \;=\; \textstyle\sum_i (\mathbf{l}_i^\mathsf{T}\mathsf{P}_i\mathbf{Q})^2.$$

Les coordonnées du point $\boldsymbol{N}$ minimisent de même le critère E, mais sont orthogonales au vecteur $\mathbf{M}$:

$$\min_{\mathbf{N},\|\mathbf{N}\|^2=1,\mathbf{N}^\mathsf{T}\mathbf{M}=0} E(\mathbf{N},\mathcal{M}).$$

Une erreur "duale" à l'erreur de reprojection. Notons $\hat{\mathbf{q}}_i = \mathsf{P}_i\mathbf{Q}$ les coordonnées des reprojections d'un point $\boldsymbol{Q}$. Le critère E se reécrit comme :

$$E(\mathbf{Q}, \mathcal{M}) \quad = \quad \sum_i (\mathbf{l}_i^\mathsf{T} \hat{\mathbf{q}}_i)^2 \quad = \quad \sum_i d^2_{A_PH}(\hat{\mathbf{q}}_i, \mathbf{l}_i),$$

où $d^2_{A_PH}(\mathbf{q}, \mathbf{l}) = (\mathbf{q}^\mathsf{T}\mathbf{l})^2$ est la distance algébrique définie par l'équation (2.19). Notons que cette distance est reliée à la distance Euclidienne par un facteur multiplicatif. La méthode de Hartley minimise donc l'erreur algébrique entre les reprojections des deux points représentant la droite et les droites image observées.

Cette méthode donne des résultats acceptables, voir §6.5. Cependant, la forme de l'erreur minimisée est différente de celle de l'erreur de reprojection basée sur une distance entre des points mesurés et une droite prédite (et non l'inverse). L'application d'un schéma de résolution de type quasi-linéaire ne permettra donc pas d'obtenir la minimisation de l'erreur de reprojection.

6.3.2 Choix de la représentation

Comme nous le soulignons dans l'introduction, une des difficultés principales concernant la manipulation de droites 3D est la représentation algébrique. La première question à se poser est donc quelle représentation choisir, étant donnés les critères présentés en §6.2. Un critère spécifique au problème de la triangulation entre en jeu. Nous cherchons une solution permettant d'initialiser la reconstruction, et un algorithme linéaire est souvent préférable. Etant donnée la forme de l'erreur de reprojection, voir équation (6.1), basée sur les reprojections de la droite, notre critère principal de choix est le suivant : nous désirons que les coordonnées des reprojections de la droite reconstruite soient linéaires en les paramètres de la représentation choisie. Parmi les représentations de la section 6.2, seules les coordonnées de Plücker satisfont cette propriété.

6.3.3 Projection d'une droite en coordonnées de Plücker

Faugeras et Mourrain [62] exhibent une matrice de projection notée $\tilde{\mathsf{P}}$, de taille (3×6), dont les coefficients s'expriment en fonction de la matrice de projection (3×4) pour points. Cette matrice de projection permet d'écrire :

$$\hat{\mathbf{l}}_i \quad \sim \quad \tilde{\mathsf{P}}_i\mathbf{L}.$$

Ils interprètent, ainsi que Hartley et Zisserman [93], les lignes de cette matrice comme les coordonnées de Plücker de 3 droites contenant le centre de projection de la caméra et définissant une base de l'espace centrée sur cette caméra.

Nous montrons comment exprimer $\tilde{\mathsf{P}}$ simplement en fonction de $\mathsf{P} \sim (\ \bar{\mathsf{P}} \quad \mathbf{p})$, la matrice de projection (3×4) pour points. Considérons une droite avec des coordonnées de Plücker $\mathbf{L}^\mathsf{T} \sim (\ \mathbf{a}^\mathsf{T} \quad \mathbf{b}^\mathsf{T}\ )$ définie par deux points de coordonnées $\mathbf{M}^\mathsf{T} \sim (\ \bar{\mathbf{M}}^\mathsf{T} \quad m)$ et $\mathbf{N}^\mathsf{T} \sim (\ \bar{\mathbf{N}}^\mathsf{T} \quad n)$, projetés en $\boldsymbol{m}$ et $\boldsymbol{n}$ respectivement par la matrice de projection P. Les coordonnées de la droite image correspondante sont donc $\mathbf{l} \sim \mathbf{m} \wedge \mathbf{n}$. Si l'on remplace $\boldsymbol{m}$ et $\boldsymbol{n}$ par les projections des points $\boldsymbol{M}$ et $\boldsymbol{N}$ respectivement, on obtient $\mathbf{l} \sim (\mathsf{P}\mathbf{M}) \wedge (\mathsf{P}\mathbf{N})$. Développons cette expression :

$$\begin{aligned} \mathbf{l} \quad &\sim \quad (\bar{\mathsf{P}}\bar{\mathbf{M}} + m\mathbf{p}) \wedge (\bar{\mathsf{P}}\bar{\mathbf{N}} + n\mathbf{p}) \\ &\sim \quad (\bar{\mathsf{P}}\bar{\mathbf{M}}) \wedge (\bar{\mathsf{P}}\bar{\mathbf{N}}) + m\mathbf{p} \wedge (\bar{\mathsf{P}}\bar{\mathbf{N}}) - n\mathbf{p} \wedge (\bar{\mathsf{P}}\bar{\mathbf{M}}). \end{aligned}$$

En utilisant l'équation (2.2) : $(\mathsf{S}\mathbf{x}) \wedge (\mathsf{S}\mathbf{y}) = \mathsf{S}^\star(\mathbf{x} \wedge \mathbf{y})$, où $\mathsf{S}^\star$ est la matrice des cofacteurs[3] de S :

$$\mathbf{l} \quad \sim \quad \bar{\mathsf{P}}^\star(\bar{\mathbf{M}} \wedge \bar{\mathbf{N}}) + [\mathbf{p}]_\wedge \bar{\mathsf{P}}(m\bar{\mathbf{N}} - n\bar{\mathbf{M}}),$$

où l'on identifie la définition (2.54) des coordonnées de Plücker, ce qui conduit à :

$$\mathbf{l} \quad \sim \quad \tilde{\mathsf{P}}\mathbf{L} \quad \text{avec} \quad \tilde{\mathsf{P}}_{(3\times 6)} \quad \sim \quad (\ \bar{\mathsf{P}}^\star \quad [\mathbf{p}]_\times \bar{\mathsf{P}}\ ). \tag{6.3}$$

Le tableau 6.2 résume les formes que prend cette matrice de projection pour différents niveaux de calibrage de la caméra.

[3] $\mathsf{S}^\star$ est toujours définie. Elle est donnée par $\mathsf{S}^\star = \det(\mathsf{S})\mathsf{S}^{-\mathsf{T}}$ si S est inversible. Ses propriétés sont données en §2.3.

espace	projection de points		projection de droites	
Projectif	$(\ \bar{\mathsf{P}}$	$\mathbf{p})$	$(\bar{\mathsf{P}}^\star$	$[\mathbf{p}]_\times\bar{\mathsf{P}}\ )$
Affine	$(\ \bar{\mathsf{A}}$	$\mathbf{a})$	$(\bar{\mathsf{A}}^\star$	$[\mathbf{a}]_\times\bar{\mathsf{A}}\ )$
Métrique	$(\ s\mathsf{R}$	$\mathbf{t})$	$(s^2\mathsf{R}$	$[\mathbf{t}]_\times\mathsf{R}\ )$
Euclidien	$(\ \mathsf{R}$	$\mathbf{t})$	$(\mathsf{R}$	$[\mathbf{t}]_\times\mathsf{R}\ )$

TAB. 6.2 – Matrices de projection (3×6) pour coordonnées de Plücker pour différents niveaux de calibrage de la caméra. La notation $\mathsf{S}^\star$ désigne la matrice des cofacteurs de la matrice S.

6.3.4 Algorithme linéaire

Nous décrivons un algorithme linéaire trivial. Ayant choisi notre représentation, les coordonnées de Plücker, et ayant établi une expression linéaire pour la projection perspective, nous pouvons reécrire l'erreur de reprojection en formulant explicitement les coordonnées des droites reprojetées :

$$\mathcal{L}(\mathbf{L},\mathcal{M}) \quad = \quad \sum_{i=1}^{n}\left(d^2_{PH}(\mathbf{x}_i,\tilde{\mathsf{P}}_i\mathbf{L}) + d^2_{PH}(\mathbf{y}_i,\tilde{\mathsf{P}}_i\mathbf{L})\right).$$

Cette fonction est une somme de carrés de termes non-linéaires. Chaque terme est basé sur la distance Euclidienne point-droite 2D, définie par l'équation (2.14). Si l'on suppose que l'équation de chaque point $\boldsymbol{q}$ est normalisée telle que $q_3 = 1$, alors cette distance s'écrit :

$$d^2_{PH}(\mathbf{q},\mathbf{l}) \quad = \quad \frac{(\mathbf{q}^\mathsf{T}\mathbf{l})^2}{l_1^2+l_2^2}.$$

Le dénominateur de cette distance est la cause de la non-linéarité des termes. Si on l'ignore, on retrouve la distance algébrique (2.19), biaisée par rapport à la distance Euclidienne, mais linéaire en les coordonnées de la droite prédite. Définissons le facteur de biais par $w^2 = \frac{1}{l_1^2+l_2^2}$. Il relie les distances algébrique et Euclidienne par l'équation (2.22) :

$$d^2_{A_PH}(\mathbf{q},\mathbf{l}) \quad = \quad w^2\, d^2_{PH}(\mathbf{q},\mathbf{l}).$$

On peut définir une fonction de coût basée sur une somme de carrés de termes linéaires en remplaçant la distance Euclidienne par la distance algébrique dans l'erreur de reprojection. Nous notons cette fonction $\mathcal{B}$ pour "biaisée" :

$$\begin{aligned}\mathcal{B}(\mathbf{L},\mathcal{M}) \quad &= \quad \sum_{i=1}^{n}\left(d^2_{A_PH}(\mathbf{x}_i,\tilde{\mathsf{P}}_i\mathbf{L}) + d^2_{A_PH}(\mathbf{y}_i,\tilde{\mathsf{P}}_i\mathbf{L})\right)\\ &= \quad \sum_{i=1}^{n}\left((\mathbf{x}_i^\mathsf{T}\tilde{\mathsf{P}}_i\mathbf{L})^2 + (\mathbf{y}_i^\mathsf{T}\tilde{\mathsf{P}}_i\mathbf{L})^2\right),\end{aligned}$$

ou sous forme matricielle :

$$\mathcal{B}(\mathbf{L},\mathcal{M}) \quad = \quad \|\mathsf{A}\mathbf{L}\|^2 \quad \text{avec} \quad \mathsf{A}_{(2n\times 6)} \quad = \quad \begin{pmatrix}\cdots\\ \mathbf{x}_i^\mathsf{T}\tilde{\mathsf{P}}_i\\ \mathbf{y}_i^\mathsf{T}\tilde{\mathsf{P}}_i\\ \cdots\end{pmatrix}. \tag{6.4}$$

Comme $\mathbf{L}$ est un vecteur de coordonnées homogènes, nous fixons son échelle avec la contrainte $\|\mathbf{L}\|^2_2 = 1$. Le vecteur $\mathbf{L}$ qui minimise $\mathcal{B}(\mathbf{L},\mathcal{M})$ est donné par le vecteur singulier de la matrice A associé à sa plus petite valeur singulière.

Cette méthode permet d'obtenir une initialisation de la structure. Les expérimentations montrent qu'elle donne de mauvais résultats. Ceci provient du fait que le résultat est biaisé pour deux raisons. La première est

due à la fonction de coût considérée, basée sur une distance algébrique. La deuxième provient du fait que la contrainte de Plücker est ignorée par la méthode : en général, en présence de bruit dans les données, le vecteur-6 retrouvé ne satisfera pas cette contrainte. On peut faire l'analogie avec l'algorithme des huit points pour le calcul de la matrice fondamentale : la matrice (3×3) retrouvée de manière linéaire ne satisfait pas la contrainte de rang deux, et doit être corrigée en ce sens. Le vecteur-6 calculé de manière linéaire est donc corrigé, ce qui est la deuxième source de biais de la méthode. L'algorithme est donné dans le tableau 6.3. Notre méthode de correction est introduite dans la section suivante.

1. *Solution linéaire* : former le système linéaire représenté par A d'après l'équation (6.4) et déterminer $\mathbf{L}$ comme le vecteur singulier associé à la plus petite valeur singulière de A ;
2. *Correction de Plücker* : appliquer la correction de Plücker décrite dans le tableau C.1 sur le vecteur $\mathbf{L}$.

TAB. 6.3 – Algorithme de triangulation linéaire LIN.

6.3.5 Correction de Plücker

Soit $\mathbf{L}^\mathsf{T} \sim (\ \mathbf{a}^\mathsf{T} \quad \mathbf{b}^\mathsf{T}\ )$ un vecteur-6 ne satisfaisant pas nécessairement la contrainte de Plücker (2.55), c'est à dire que $\mathbf{a}^\mathsf{T}\mathbf{b}$ peut ne pas être nul. Nous cherchons un vecteur $\hat{\mathbf{L}}^\mathsf{T} \sim (\ \mathbf{u}^\mathsf{T} \quad \mathbf{v}^\mathsf{T}\ )$ tel que $\mathbf{u}^\mathsf{T}\mathbf{v} = 0$ et tel qu'un certain *critère de correction* soit minimisé. Pour choisir ce critère de correction, utilisons l'analogie avec l'algorithme des huit points. Soit F la matrice estimée de manière linéaire. La correction du rang est effectuée en trouvant la matrice de rang deux $\hat{\mathsf{F}}$ telle que la distance entre F et $\hat{\mathsf{F}}$ soit minimisée. Cette distance s'exprime à l'aide de la norme de Frobenius, et le problème de la correction de rang se formule comme :

$$\min_{\hat{\mathsf{F}}, \det \hat{\mathsf{F}}=0} \|\mathsf{F} - \hat{\mathsf{F}}\|^2.$$

Une solution élégante basée sur la décomposition de F en valeurs singulières est possible, voir [86, 233].

Dans le cas des coordonnées de Plücker, nous formulons le problème similairement par :

$$\min_{\hat{\mathbf{L}}, \mathbf{u}^\mathsf{T}\mathbf{v}=0} \mathcal{O} \quad \text{avec} \quad \mathcal{O} \ = \ \|\mathbf{L} - \hat{\mathbf{L}}\|^2. \tag{6.5}$$

Ce problème a une formulation simple et concise, mais il n'est pas trivial. Notre solution est décrite dans l'annexe C.

6.3.6 Algorithme quasi-linéaire 1

Nous proposons un algorithme de triangulation quasi-linéaire basé sur le schéma décrit au chapitre 2, §2.6.2, permettant de corriger la première source de biais de l'algorithme linéaire, c'est à dire l'approximation de la distance Euclidienne par une distance algébrique. L'idée est de repondérer itérativement le système d'équations utilisé. En effet, les distances Euclidienne et algébrique sont reliées par un facteur de biais défini par l'équation (2.23) et dont la valeur dépend de la droite reprojetée. L'erreur de reprojection (6.1) et l'erreur biaisée (6.4) sont donc reliées par un ensemble de facteurs de biais, un pour chaque image de la droite. Le tableau 6.4 résume l'algorithme proposé.

Le système linéaire initial est formé puis résolu au sens des moindres carrés comme pour la méthode linéaire. Le vecteur-6 calculé est noté $\mathbf{L}_0$. Il est corrigé de manière à satisfaire la contrainte de Plücker. Basé sur cette estimation, les facteurs de biais peuvent être calculés et utilisés pour repondérer le système linéaire. Le procédé est itéré pour calculer des estimations successives $\mathbf{L}_k$ des coordonnées de Plücker de la droite 3D minimisant l'erreur de reprojection (k est le compteur d'itération). La convergence est déterminée en seuillant la différence entre deux erreurs consécutives. Elle est atteinte en typiquement 3 ou 4 itérations.

Nos résultats expérimentaux montrent que cet algorithme n'améliore que très peu les résultats obtenus par l'algorithme linéaire. En effet, seule une source de biais est traitée. Il s'avère que la source principale du biais est la non prise en compte de la contrainte de Plücker dans les équations. Des effets similaires sont décrits par Zhang [233] à propos d'une méthode semblable pour l'estimation de la matrice fondamentale.

1. *Initialisation* : former le système linéaire $\mathsf{A} = \mathsf{A}_0$ d'après l'équation (6.4), initialiser $k = 0$;
2. *Estimation* : calculer $\mathbf{L}_k$ tel que $\min_{\mathbf{L}_k, \|\mathbf{L}_k\|^2=1} \|\mathsf{A}_k \mathbf{L}_k\|^2$ puis appliquer la correction de Plücker, voir l'algorithme du tableau C.1 ;
3. *Correction du biais de l'erreur algébrique* : former la matrice A_{k+1} en repondérant A_k par les facteurs de biais calculés d'après l'équation (2.23) ;
4. *Itération* : $k \leftarrow k + 1$, itérer les étapes 2 et 3 jusqu'à convergence.

TAB. 6.4 – Algorithme de reconstruction quasi-linéaire naïf "QLIN1". Seul le biais de l'algorithme linéaire dû à la mesure de distance algébrique est pris en compte.

6.3.7 Algorithme quasi-linéaire 2

Nous proposons un algorithme quasi-linéaire permettant de corriger les deux sources de biais de la méthode linéaire. Reprenons l'algorithme quasi-linéaire précédent. Cet algorithme corrige le biais lié à l'erreur algébrique. Nous proposons de linéariser la contrainte bilinéaire de Plücker et de l'incorporer au système linéaire résolu à chaque itération. Nous pouvons attendre de bons résultats d'un tel algorithme. En effet, les deux sources de biais sont traitées et la contrainte de Plücker étant bilinéaire, elle devrait être assez bien approchée localement par un terme linéaire. L'algorithme est résumé dans le tableau 6.5.

Considérons la contrainte de Plücker (2.55) et reécrivons-la comme suit :

$$\mathcal{K}(\mathbf{L}) \;=\; \mathbf{L}^\mathsf{T} \mathsf{Q} \mathbf{L} \quad \text{avec} \quad \mathsf{Q}_{(6\times 6)} \;=\; \begin{pmatrix} 0 & \mathrm{I} \\ \mathrm{I} & 0 \end{pmatrix}.$$

Notons que Q représente la quadrique de Klein dans $\mathbb{P}^5$, voir le chapitre 2, §2.12. Développons cette contrainte au premier ordre autour de l'estimation courante $\mathbf{L}_k$, en supposant que l'estimation suivante n'est pas "trop loin" ($\mathbf{L}_{k+1} = \mathbf{L}_k + \boldsymbol{\delta}$) :

$$\begin{aligned} \mathcal{K}(\mathbf{L}_{k+1}) &= \mathcal{K}(\mathbf{L}_k + \boldsymbol{\delta}) \\ &= \underbrace{\mathcal{K}(\mathbf{L}_k) + \left.\frac{\partial \mathcal{K}(\mathbf{L})}{\partial \mathbf{L}}\right|_{\mathbf{L}_k} \cdot \boldsymbol{\delta}}_{\mathcal{K}_k(\mathbf{L}_{k+1})} + \ldots, \end{aligned}$$

où $\mathcal{K}_k$ dénote la contrainte de Plücker linéarisée au voisinage de $\mathbf{L}_k$. Comme $\mathcal{K}(\mathbf{L}_k) = 0$ (l'estimation à l'itération k satisfait la contrainte de Plücker) et $\left.\frac{\partial \mathcal{K}(\mathbf{L})}{\partial \mathbf{L}}\right|_{\mathbf{L}_k} \cdot \mathbf{L}_k = 0$:

$$\begin{aligned} \mathcal{K}_k(\mathbf{L}_{k+1}) &= \mathcal{K}(\mathbf{L}_k) + \left.\frac{\partial \mathcal{K}(\mathbf{L})}{\partial \mathbf{L}}\right|_{\mathbf{L}_k} \cdot \boldsymbol{\delta} \\ &= \left.\frac{\partial \mathcal{K}(\mathbf{L})}{\partial \mathbf{L}}\right|_{\mathbf{L}_k} \cdot \boldsymbol{\delta} + \underbrace{\left.\frac{\partial \mathcal{K}(\mathbf{L})}{\partial \mathbf{L}}\right|_{\mathbf{L}_k} \cdot \mathbf{L}_k}_{0} \\ &= \left.\frac{\partial \mathcal{K}(\mathbf{L})}{\partial \mathbf{L}}\right|_{\mathbf{L}_k} \cdot \mathbf{L}_{k+1}, \end{aligned}$$

d'où l'expression de la contrainte linéarisée :

$$\mathcal{K}_k(\mathbf{L}_{k+1}) \;=\; \mathbf{L}_k^\mathsf{T} \mathsf{Q} \mathbf{L}_{k+1} \;=\; 0. \tag{6.6}$$

Nous adaptons le schéma d'optimisation linéaire sous contraintes linéaires décrit en §2.6.4.2 pour minimiser l'erreur de moindres carrés linéaires sous les contraintes $\mathcal{K}_k(\mathbf{L}_{k+1}) = 0$ et $\|\mathbf{L}_{k+1}\|^2 = 1$. L'idée est de calculer un sous-espace de paramètres de dimension 5 où la contrainte de Plücker linéarisée est satisfaite. Le problème

d'optimisation est résolu dans cet espace, puis le vecteur solution $\boldsymbol{\gamma}$ sera transféré en coordonnées de Plücker. Considérons l'expression (6.6) de la contrainte de Plücker linéarisée. Les vecteurs $\mathbf{L}_{k+1}$ satisfaisant cette contrainte sont orthogonaux au vecteur $\mathsf{Q}\mathbf{L}_k$ et sont donc dans l'espace nul associé à $\mathbf{L}_k^\mathsf{T}\mathsf{Q}$. Cet espace nul peut être calculé par exemple par la décomposition en valeurs singulières suivante :

$$\mathbf{L}_k^\mathsf{T}\mathsf{Q} \quad \sim \quad \mathbf{u}^\mathsf{T}\mathrm{diag}(1,0,0,0,0,0)(\mathbf{v}_{(6\times1)} \;\; \bar{\mathsf{V}}_{(6\times5)})^\mathsf{T}.$$

Les colonnes de la matrice $\bar{\mathsf{V}}$ forment une base orthogonale de cet espace :

$$\mathbf{L}_k^\mathsf{T}\mathsf{Q}\bar{\mathsf{V}} \quad = \quad \mathbf{0}.$$

Définissons $\mathbf{L}_{k+1} = \bar{\mathsf{V}}\boldsymbol{\gamma}$, il est trivial de voir qu'alors $\mathbf{L}_{k+1}$ satisfait la contrainte de Plücker linéarisée :

$$\mathbf{L}_k^\mathsf{T}\mathsf{Q}\mathbf{L}_{k+1} \quad = \quad \mathbf{L}_k^\mathsf{T}\mathsf{Q}\bar{\mathsf{V}}\boldsymbol{\gamma} \quad = \quad 0.$$

Substituons cette expression de $\mathbf{L}_{k+1}$ dans la forme matricielle de la fonction de coût (6.4), nous obtenons :

$$\|\mathsf{A}\mathbf{L}_{k+1}\|^2 \quad = \quad \|\mathsf{A}\bar{\mathsf{V}}\boldsymbol{\gamma}\|^2.$$

Le vecteur singulier associé à la plus petite valeur singulière de la matrice $\mathsf{A}\bar{\mathsf{V}}$ donne la solution pour $\boldsymbol{\gamma}$ minimisant l'erreur $\mathcal{B}(\mathbf{L}_{k+1}, \mathcal{M})$, sous la contrainte $\|\boldsymbol{\gamma}\|^2 = 1$. Notons que comme $\bar{\mathsf{V}}$ a des colonnes orthonormales, $\|\mathbf{L}_{k+1}\|^2 = \|\boldsymbol{\gamma}\|^2 = 1$.

1. *Initialisation* : former le système linéaire $\mathsf{A} = \mathsf{A}_0$ d'après l'équation (6.4), calculer $\mathbf{L}_0$ par minimisation de $\|\mathbf{A}\mathbf{L}_0\|^2$, voir l'algorithme du tableau 6.3 et par application de la correction de Plücker, voir l'algorithme du tableau C.1. Initialiser $k = 0$;
2. *Linéarisation de la contrainte de Plücker* : calculer la décomposition en valeurs singulières $\mathbf{L}_k^\mathsf{T}\mathsf{Q} \sim \mathbf{u}^\mathsf{T}\mathrm{diag}(1,0,0,0,0,0)(\mathbf{v}_{(6\times1)} \;\; \bar{\mathsf{V}}_{(6\times5)})^\mathsf{T}$;
3. *Estimation* : calculer $\mathbf{L}_{k+1} = \bar{\mathsf{V}}\boldsymbol{\gamma}$ avec $\min_{\boldsymbol{\gamma},\|\boldsymbol{\gamma}\|^2=1} \|\mathsf{A}\bar{\mathsf{V}}\boldsymbol{\gamma}\|^2$ puis appliquer la correction de Plücker, algorithme du tableau C.1 ;
4. *Correction du biais de l'erreur algébrique* : repondérer le système linéaire A par les poids calculés avec l'équation (2.23) ;
5. *Itération* : $k \leftarrow k + 1$, itérer les étapes 2, 3 et 4 jusqu'à convergence.

TAB. 6.5 – Algorithme de reconstruction quasi-linéaire "QLIN2". Cet algorithme prend en compte les deux sources de biais de l'algorithme linéaire : la mesure d'erreur algébrique et la contrainte de Plücker.

6.3.8 Algorithme non-linéaire

Nous avons proposé des solutions de reconstruction linéaires et quasi-linéaires. Ces solutions exploitent la structure du problème, c'est à dire le lien entre l'erreur de reprojection et l'erreur algébrique. Elles ne permettent pas d'utiliser n'importe quelle représentation des droites 3D.

Nous proposons d'utiliser une technique d'optimisation non-linéaire plus générale afin de permettre l'utilisation de paramétrisations non-linéaires des droites 3D. Nous utilisons l'algorithme de Levenberg-Marquardt, présenté au chapitre 3, §3.6.2, pour une minimisation directe de l'erreur de reprojection. La solution initiale est fournie par les algorithmes précédemment proposés. La paramétrisation des droites 3D est basée sur leur représentation orthonormale, que nous présentons à la section suivante, dans le contexte plus général de l'ajustement de faisceaux. Cet algorithme de reconstruction est nommé "NLIN". L'intérêt principal est de comparer les résultats obtenus par QLIN2 et NLIN.

6.4 Ajustement de faisceaux

L'ajustement de faisceaux est essentiellement la minimisation de l'erreur de reprojection sur l'ensemble du modèle 3D, c'est à dire sur les paramètres des caméras et des droites. Nous renvoyons au chapitre 3 pour plus de détails et la justification de l'erreur de reprojection comme fonction de coût. L'ajustement de faisceaux est un processus d'optimisation non-linéaire et requiert donc une solution initiale du problème. Des méthodes pour déterminer une solution initiale ont fait l'objet de la section précédente. L'ajustement de faisceaux requiert de plus une paramétrisation du problème, c'est à dire un ensemble de paramètres à partir desquels le modèle reconstruit peut être exprimé, afin de pouvoir calculer la valeur de la fonction de coût. L'optimisation non-linéaire agit sur cette paramétrisation. La paramétrisation est divisée en celle du mouvement des caméras et celle de la structure, c'est à dire des droites 3D. Pour la paramétrisation des caméras, nous renvoyons au chapitre 3, §3.5.5. Nous nous concentrons sur la paramétrisation des droites 3D. Différentes représentations algébriques des droites 3D ont été présentées et discutées en §6.2. Produire une "bonne" paramétrisation des droites 3D est difficile car, contrairement aux points ou plans, il n'existe pas de représentation et donc de paramétrisation "naturelle".

Nous rappelons l'expression de l'erreur de reprojection, donnée au chapitre 3, §3.3.2. En reprenant les notations de la section 6.3, pour un ensemble de droites $\mathcal{S} = \{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_m\}$ et un ensemble de caméras $\mathcal{M} = \{\mathsf{P}_1, \ldots, \mathsf{P}_n\}$, elle s'exprime comme la somme des erreurs de reprojection individuelles :

$$\begin{aligned} \mathcal{L}(\mathcal{S}, \mathcal{M}) &= \sum_{j=1}^{m} \mathcal{L}(\boldsymbol{L}_j, \mathcal{M}) \\ &= \sum_{j=1}^{m} \sum_{i=1}^{n} \left(d^2_{PH}(\mathbf{x}_{ij}, \hat{\mathbf{l}}_{ij}) + d^2_{PH}(\mathbf{y}_{ij}, \hat{\mathbf{l}}_{ij}) \right). \end{aligned} \tag{6.7}$$

Cette formulation suppose, sans perte de généralité, que toutes les droites sont visibles dans toutes les vues.

6.4.1 Etat de l'art

Il existe peu de travaux dédiés à la reconstruction de droites par ajustement de faisceaux. Le problème de la paramétrisation des droites 3D est souvent mis de côté. La plupart des auteurs utilisent une paramétrisation basée sur les représentations duales, par deux points ou par deux plans.

Hartley [92] propose un algorithme de calcul optimal du tenseur trifocal, basé sur des correspondances de droites et / ou de points sur trois vues. Il réalise un ajustement de faisceaux en utilisant un algorithme d'optimisation non-linéaire de type Levenberg-Marquardt. Il montre que la représentation des droites 3D par deux droites image est pratique dans ce cas, car la reprojection de la droite reconstruite dans la troisième image est obtenue directement en fonction des deux droites images la représentant. Cet algorithme est nommé MLE_HARTLEY. Il est comparé aux nôtres en §6.5.

Nous étendons cet algorithme au cas de vues multiples. La difficulté est que dans ce cas, on ne peut pas sélectionner a priori deux images où les droites seront représentées, car les droites ne sont pas forcement visibles dans toutes les vues, et choisir deux vues trop proches induirait l'instabilité numérique de la représentation. Cette extension requiert donc une sélection des deux meilleures images permettant de représenter une droite donnée, selon un certain critère. De plus, les positions des caméras changent au cours d'un ajustement de faisceaux. La sélection des meilleures images doit donc être dynamique.

Le critère de sélection que nous utilisons est le suivant. Il consiste à choisir la paire d'images induisant la matrice fondamentale la mieux définie possible, ce que nous mesurons par un critère de choix de modèle inspiré de celui de Torr [207]. Nous remarquons dans nos expérimentations que ce choix influence très peu la qualité de l'estimation.

6.4.2 La représentation orthonormale des droites 3D

Nous proposons une représentation orthonormale des droites 3D. Cette représentation présente des caractéristiques similaires à celles de la représentation orthonormale de la matrice fondamentale présentée au chapitre

4, §4.4. Elle est basée sur des matrices orthonormales, elle est redondante et complète (toutes les droites sont représentées) et permet une mise à jour locale avec le nombre minimum de 4 paramètres. Nous donnons une preuve de l'existence de la représentation orthonormale et un algorithme de construction à partir des coordonnées de Plücker. Nous proposons une loi de mise à jour et exhibons une expression analytique pour la matrice Jacobienne de cette mise à jour. Ces résultats sont résumés dans le tableau 6.6. La figure 6.1 illustre le principe de l'utilisation de la représentation orthonormale. Le vocabulaire suivant est employé. La *décomposition*

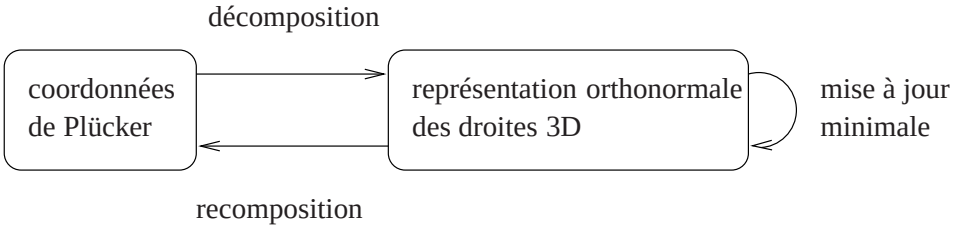


FIG. 6.1 – Principe d'utilisation de la représentation orthonormale des droites 3D pour l'optimisation non-linéaire.

désigne le passage des coordonnées de Plücker à la représentation orthonormale et la *recomposition* l'inverse.

Nous étudions les opérations de décomposition et de recomposition ainsi que leurs propriétés de consistance et de complétude. Nous donnons ensuite un schéma de mise à jour locale minimale de la représentation orthonormale et étudions ses propriétés de consistance et de complétude. Nous proposons une interprétation géométrique de la représentation orthonormale et des 4 paramètres du schéma de mise à jour proposé. Finalement, nous étudions deux familles de droites dont la représentation orthonormale n'est pas unique, et les conséquences sur l'optimisation non-linéaire. La représentation orthonormale et le schéma de mise à jour sont résumés dans le tableau 6.6.

6.4.2.1 Décomposition

Considérons les coordonnées de Plücker $\mathbf{L}^\mathsf{T} \sim \begin{pmatrix} \mathbf{a}^\mathsf{T} & \mathbf{b}^\mathsf{T} \end{pmatrix}$ et la matrice $\bar{\mathsf{C}} \sim (\mathbf{a}\ \mathbf{b})$ déjà utilisée pour l'algorithme de correction de Plücker donné en annexe C. La contrainte de Plücker (2.55) $\mathbf{a}^\mathsf{T}\mathbf{b} = 0$ est équivalente à l'orthogonalité des colonnes de $\bar{\mathsf{C}}$. Notre représentation orthonormale se construit intuitivement par normalisation des colonnes de $\bar{\mathsf{C}}$ puis ajout d'une troisième colonne afin de rendre la matrice orthonormale. Le facteur d'échelle relatif entre les deux premières colonnes est modélisé indépendamment. L'équation suivante reflète cette idée :

$$\bar{\mathsf{C}}_{(3\times2)} \sim (\mathbf{a}\ \mathbf{b}) \sim \underbrace{\begin{pmatrix} \frac{\mathbf{a}}{\|\mathbf{a}\|} & \frac{\mathbf{b}}{\|\mathbf{b}\|} & \frac{\mathbf{a}\wedge\mathbf{b}}{\|\mathbf{a}\|\|\mathbf{b}\|} \end{pmatrix}}_{\text{matrice orthonormale}} \underbrace{\begin{pmatrix} \|\mathbf{a}\| & 0 \\ 0 & \|\mathbf{b}\| \\ 0 & 0 \end{pmatrix}}_{\text{point de } \mathbb{P}^1}. \tag{6.8}$$

Les directions données par les vecteurs orthogonaux $\mathbf{a}$ et $\mathbf{b}$ sont encapsulées dans une matrice orthonormale. Seul le rapport de leurs normes est significatif, il est équivalent à un point de la droite projective $\mathbb{P}^1$ ou à une rotation 2D. Pour éviter les nombreux cas particuliers se présentant avec une décomposition "manuelle", en particulier lorsque $\|\mathbf{a}\| = 0$ ou $\|\mathbf{b}\| = 0$, nous utilisons la décomposition QR, voir chapitre 2, §2.4.2. Elle fournit un moyen simple et élégant de construction de la représentation orthonormale :

$$\bar{\mathsf{C}}_{(3\times2)} \sim \mathsf{U}_{(3\times3)}\Sigma_{(3\times2)}, \tag{6.9}$$

où U est une matrice orthonormale et Σ est une matrice triangulaire supérieure paramétrée par :

$$\Sigma \sim \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{pmatrix}.$$

L'élément $(1,2)$ de la matrice Σ est nul car sinon l'orthogonalité de $\bar{\mathsf{C}}$ et donc la contrainte de Plücker seraient violées. La matrice Σ est définie à un facteur près car elle encapsule le facteur d'échelle libre de $\bar{\mathsf{C}}$. Il existe donc $\alpha \in \mathbb{R}$, tel que $(\sigma_1 \;\; \sigma_2) \sim (\cos\alpha \;\; \sin\alpha)$. On peut donc définir une matrice de rotation 2D W par :

$$\mathsf{W} \;=\; \tfrac{1}{\sigma_1^2+\sigma_2^2}\begin{pmatrix}\sigma_1 & -\sigma_2\\ \sigma_2 & \sigma_1\end{pmatrix} \;=\; \begin{pmatrix}\cos\alpha & -\sin\alpha\\ \sin\alpha & \cos\alpha\end{pmatrix}. \tag{6.10}$$

Cette décomposition montre que toute droite 3D peut être représentée par un doublon (U,W), une matrice orthonormale de $O(3)$ et une matrice orthonormale de $SO(2)$. Nous appelons cette décomposition la *représentation orthonormale des droites 3D*. Notons que la représentation orthonormale a un degré d'ambiguïté lorsque la droite est à l'infini ou lorsqu'elle contient l'origine. Ces cas sont étudiés en §6.4.2.6.

6.4.2.2 Recomposition

Soit $(\mathsf{U},\mathsf{W}) \in O(3)\times SO(2)$ la représentation orthonormale d'une droite 3D. Les valeurs de σ_1 et σ_2 sont extraites de $\mathsf{W} \sim \left(\begin{smallmatrix} w_{11} & w_{12}\\ w_{21} & w_{22}\end{smallmatrix}\right)$, par l'équation (6.10) :

$$\sigma_1 \;=\; w_{11} \quad \text{et} \quad \sigma_2 \;=\; w_{21}.$$

Le calcul des coordonnées de Plücker est donné par la recomposition de la factorisation QR, équation (6.9) :

$$\bar{\mathsf{C}} \;\sim\; \mathsf{U}\begin{pmatrix} w_{11} & 0\\ 0 & w_{21}\\ 0 & 0\end{pmatrix},$$

ce qui donne l'*équation de recomposition* :

$$\mathbf{L}(\mathsf{U},\mathsf{W})^{\mathsf{T}} \;=\; (\;\; w_{11}\mathbf{u}_1^{\mathsf{T}} \qquad w_{21}\mathbf{u}_2^{\mathsf{T}} \;\;). \tag{6.11}$$

6.4.2.3 Consistance et complétude

Consistance. Les contraintes de consistance devant être satisfaites par un doublon $(\mathsf{U},\mathsf{W}) \in O(3)\times SO(2)$ sont l'orthonormalité des matrices U et W :

$$\mathsf{U}^{\mathsf{T}}\mathsf{U} \;=\; \mathrm{I}_{(3\times 3)}, \quad \mathsf{W}^{\mathsf{T}}\mathsf{W} \;=\; \mathrm{I}_{(2\times 2)} \quad \text{et} \quad \det(\mathsf{W}) = 1, \tag{6.12}$$

Si ces contraintes sont violées, la loi de recomposition (6.11) ne garantit pas que le vecteur-6 retrouvé représente des coordonnées de Plücker. Notons que les contraintes de consistance (6.12) ne laissent que 4 degrés de liberté au doublon (U,W), correspondant aux 4 degrés de liberté physiques d'une droite 3D.

Complétude. La représentation orthonormale est complète dans le sens où toute droite 3D peut s'écrire en coordonnées de Plücker, et toutes coordonnées de Plücker peuvent être décomposées comme proposé ci-dessus.

6.4.2.4 Optimisation minimale

Nous proposons un schéma de mise à jour locale et minimale puis donnons une expression analytique de la matrice Jacobienne correspondante. Nous étudions ensuite ses propriétés de consistance et de complétude.

L'avantage principal des représentations orthonormales est qu'une mise à jour locale est possible, avec le nombre minimum de paramètres, sans degré de liberté de jauge ou contrainte interne et ceci de manière simple. Soit $(\mathsf{U}_k,\mathsf{W}_k)$ la représentation orthonormale d'une droite à l'itération k d'une estimation non-linéaire. La mise à jour locale est effectuée par :

$$\begin{cases} \mathsf{U}_{k+1} \;=\; \mathsf{U}_k\mathsf{R}_{3D}(\mathbf{u}_k)\\ \mathsf{W}_{k+1} \;=\; \mathsf{W}_k\mathsf{R}_{2D}(w_k), \end{cases} \tag{6.13}$$

où $\mathbf{u}_k$ est un vecteur-3 et w_k un scalaire. Nous renvoyons au chapitre 2, §2.7.3 pour la justification de ces mises à jour de matrices orthonormales. Les paramètres de mise à jour sont encapsulés dans un vecteur-4 :

$$\mathbf{x}_k^\mathsf{T} \quad = \quad (\ \mathbf{u}_k^\mathsf{T} \quad w_k). \tag{6.14}$$

D'après l'équation de recomposition (6.11) et le schéma de mise à jour (6.13), nous obtenons l'expression suivante pour $\mathbf{L}_{k+1}$:

$$\mathbf{L}_{k+1} \quad = \quad \mathbf{L}(\mathsf{U}_k\mathsf{R}_{3D}(\mathbf{u}_k), \mathsf{W}_k\mathsf{R}_{2D}(w_k)). \tag{6.15}$$

Consistance. Le schéma de mise à jour (6.13) est consistant si les contraintes de consistance (6.12) de la représentation orthonormale sont garanties sur $(\mathsf{U}_{k+1}, \mathsf{W}_{k+1})$. Concernant U, il est trivial de voir que si $\mathsf{U}_k \in O(3)$, comme $\mathsf{R}_{3D}(\mathbf{u}_k) \in SO(3)$, alors $\mathsf{U}_k\mathsf{R}_{3D}(\mathbf{u}_k) = \mathsf{U}_{k+1} \in O(3)$. Concernant W, on peut faire la même vérification en dimension 2 : il est trivial de voir que si $\mathsf{W}_k \in SO(2)$, comme $\mathsf{R}_{2D}(w) \in SO(2)$, alors $\mathsf{W}_k\mathsf{R}_{2D}(w) = \mathsf{W}_{k+1} \in SO(2)$. Le schéma de mise à jour préserve donc les contraintes de consistance.

Complétude. Le schéma de mise à jour (6.13) est complet si toute droite 3D peut être atteinte à partir de toute initialisation. Concernant U, la question suivante se pose. Cette matrice orthonormale peut avoir un déterminant positif ou négatif, mais sa mise à jour ne peut changer le signe de son déterminant :

$$\text{signe}(\det(\mathsf{U}_{k+1})) \quad = \quad \text{signe}(\det(\mathsf{U}_k)).$$

Ceci n'est pas un problème car pour toute représentation orthonormale, on peut changer le signe de U (ce qui change le signe de son déterminant : $\det(\mathsf{U}) = -\det(-\mathsf{U})$), sans changer la droite 3D représentée, d'après l'équation de recomposition (6.11) :

$$\mathbf{L}(\mathsf{U}, \mathsf{W}) \quad \sim \quad \mathbf{L}(-\mathsf{U}, \mathsf{W}).$$

Le schéma de mise à jour permet donc d'atteindre toutes les droites 3D quelque soit l'initialisation.

La matrice Jacobienne $\mathsf{A}^\perp$. L'erreur de reprojection, minimisée durant l'ajustement de faisceaux, se calcule à partir des reprojections des droites, lesquelles s'expriment facilement à partir des coordonnées de Plücker, voir la matrice de projection (3×6), équation (6.3). Lors de l'utilisation de la représentation orthonormale, l'optimisation est conduite sur les paramètres de mise à jour encapsulés par le vecteur-4 $\mathbf{x}$, défini par l'équation (6.14). Nous analysons l'expression de la matrice Jacobienne exprimant localement les variations des coordonnées de Plücker en fonction des variations des paramètres de mise à jour. Cette matrice est de taille (6×4). Nous notons $\mathsf{A}^\perp$ son expression évaluée en $\mathbf{x} = \mathbf{0}$, c'est à dire autour de l'estimation courante (voir le chapitre 3, §3.5.2) :

$$\mathsf{A}^\perp \quad = \quad \left.\frac{\partial \mathbf{L}_{k+1}}{\partial \mathbf{x}_k}\right|_{\mathbf{x}_k=\mathbf{0}} \quad = \quad \left(\mathbf{a}_{u_{k,1}} \ \mathbf{a}_{u_{k,2}} \ \mathbf{a}_{u_{k,3}} \ \mathbf{a}_{w_k}\right),$$

où $\mathbf{a}_\alpha = \left.\frac{\partial \mathbf{L}_{k+1}}{\partial \alpha}\right|_{\mathbf{x}_k=\mathbf{0}}$ désigne le vecteur gradient par rapport à une variable $\alpha \in \{u_{k,1}, u_{k,2}, u_{k,3}, w_k\}$. D'après l'équation (6.15) :

$$\mathbf{a}_\alpha \quad = \quad \left.\frac{\partial \mathbf{L}(\mathsf{U}_k\mathsf{R}_{3D}(\mathbf{u}_k), \mathsf{W}_k\mathsf{R}_{2D}(w_k))}{\partial \alpha}\right|_{\mathbf{x}_k=\mathbf{0}}.$$

La fonction $\mathbf{L}(\mathsf{U}, \mathsf{W})$ impliquant un produit entre les coefficients de U et W, nous obtenons :

$$\mathbf{a}_\alpha \quad = \quad \mathbf{L}\left(\mathsf{U}_k \left.\frac{\partial \mathsf{R}_{3D}(\mathbf{u}_k)}{\partial \alpha}\right|_{\mathbf{x}_k=\mathbf{0}}, \mathsf{W}_k\mathsf{R}_{2D}(w_k)|_{\mathbf{x}_k=\mathbf{0}}\right) + \mathbf{L}\left(\mathsf{U}_k\mathsf{R}_{3D}(\mathbf{u}_k)|_{\mathbf{x}_k=\mathbf{0}}, \mathsf{W}_k \left.\frac{\partial \mathsf{R}_{2D}(w_k)}{\partial \alpha}\right|_{\mathbf{x}_k=\mathbf{0}}\right).$$

Comme indiqué au chapitre 2, §2.7.2, $\mathsf{R}_{2D}(w_k)|_{\mathbf{x}_k=\mathbf{0}} = \mathsf{R}_{2D}(0) = \mathrm{I}_{(2\times2)}$ et $\mathsf{R}_{3D}(\mathbf{u}_k)|_{\mathbf{x}_k=\mathbf{0}} = \mathsf{R}_{3D}(\mathbf{0}) = \mathrm{I}_{(3\times3)}$:

$$\mathbf{a}_\alpha \quad = \quad \mathbf{L}\left(\mathsf{U}_k \left.\frac{\partial \mathsf{R}_{3D}(\mathbf{u}_k)}{\partial \alpha}\right|_{\mathbf{x}_k=\mathbf{0}}, \mathsf{W}_k\right) + \mathbf{L}\left(\mathsf{U}_k, \mathsf{W}_k \left.\frac{\partial \mathsf{R}_{2D}(w_k)}{\partial \alpha}\right|_{\mathbf{x}_k=\mathbf{0}}\right).$$

Substituons $\alpha = u_{k,1}$ dans cette expression. Dans ce cas, $\mathsf{U}_k \frac{\partial \mathsf{R}_{3D}(\mathbf{u}_k)}{\partial u_{k,1}}\Big|_{\mathbf{x}_k=\mathbf{0}} = (\mathbf{0} \;\; \mathbf{u}_3 \;\; -\mathbf{u}_2)$ (voir le chapitre 2, §2.7.3) et $\frac{\partial \mathsf{R}_{2D}(w_k)}{\partial u_{k,1}} = 0$, ce qui conduit à :

$$\mathbf{a}_{u_{k,1}}^\mathsf{T} \;=\; \mathbf{L}\left((\mathbf{0} \;\; \mathbf{u}_3 \;\; -\mathbf{u}_2), \mathsf{W}_k\right)^\mathsf{T} \;=\; \begin{pmatrix} \mathbf{0}_{(3\times1)}^\mathsf{T} & w_{21}\mathbf{u}_3^\mathsf{T} \end{pmatrix}.$$

De manière similaire, nous obtenons les expressions des autres vecteurs gradient :

$$\begin{array}{rcl} \mathbf{a}_{u_{k,2}}^\mathsf{T} & = & \begin{pmatrix} -w_{11}\mathbf{u}_3^\mathsf{T} & \mathbf{0}_{(3\times1)}^\mathsf{T} \end{pmatrix} \\ \mathbf{a}_{u_{k,3}}^\mathsf{T} & = & \begin{pmatrix} w_{11}\mathbf{u}_2^\mathsf{T} & -w_{21}\mathbf{u}_1^\mathsf{T} \end{pmatrix}, \\ \mathbf{a}_{w_k}^\mathsf{T} & = & \begin{pmatrix} -w_{21}\mathbf{u}_1^\mathsf{T} & w_{11}\mathbf{u}_2^\mathsf{T} \end{pmatrix} \end{array}$$

ce qui donne l'expression de la matrice Jacobienne $\mathsf{A}^\perp$:

$$\mathsf{A}_{(6\times4)}^\perp \;=\; \begin{pmatrix} \mathbf{0}_{(3\times1)} & -w_{11}\mathbf{u}_3 & w_{11}\mathbf{u}_2 & -w_{21}\mathbf{u}_1 \\ \\ w_{21}\mathbf{u}_3 & \mathbf{0}_{(3\times1)} & -w_{21}\mathbf{u}_1 & w_{11}\mathbf{u}_2 \\ \\ \end{pmatrix} \tag{6.16}$$

Décomposition : des coordonnées de Plücker $\mathbf{L}^\mathsf{T} \sim \begin{pmatrix} \mathbf{a}^\mathsf{T} & \mathbf{b}^\mathsf{T} \end{pmatrix}$ à la représentation orthonormale $(\mathsf{U}, \mathsf{W}) \in O(3) \times SO(2)$:
▷ calculer la factorisation QR $(\mathbf{a} \;\; \mathbf{b}) \sim \mathsf{U}\Sigma$;
▷ former $\mathsf{W} = \frac{1}{\sigma_1^2+\sigma_2^2}\begin{pmatrix} \sigma_1 & -\sigma_2 \\ \sigma_2 & \sigma_1 \end{pmatrix}$, où $\Sigma = \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{pmatrix}$.
Recomposition : de la représentation orthonormale (U, W) aux coordonnées de Plücker $\mathbf{L}$:
▷ $\mathbf{L}^\mathsf{T} \sim \begin{pmatrix} w_{11}\mathbf{u}_1^\mathsf{T} & w_{21}\mathbf{u}_2^\mathsf{T} \end{pmatrix}$.
Mise à jour locale minimale (k est le compteur d'itérations) :
▷ *Paramètres d'optimisation* : $\mathbf{x}_k^\mathsf{T} = (\underbrace{u_{k,1} \;\; u_{k,2} \;\; u_{k,3}}_{\mathbf{u}_k} \;\; w_k)$;
▷ *Loi de mise à jour* (voir le chapitre 2 §2.7.3) :
- $\mathsf{U}_{k+1} = \mathsf{U}_k \mathsf{R}_{3D}(\mathbf{u}_k)$,
- $\mathsf{W}_{k+1} = \mathsf{W}_k \mathsf{R}_{2D}(w_k)$;

▷ *Matrice Jacobienne associée* : $\mathsf{A}_{(6\times4)}^\perp = \begin{pmatrix} \mathbf{0}_{(3\times1)} & -w_{11}\mathbf{u}_3 & w_{11}\mathbf{u}_2 & -w_{21}\mathbf{u}_1 \\ \\ w_{21}\mathbf{u}_3 & \mathbf{0}_{(3\times1)} & -w_{21}\mathbf{u}_1 & w_{11}\mathbf{u}_2 \\ \\ \end{pmatrix}$.

TAB. 6.6 – La *représentation orthonormale des droites 3D*. Ce tableau résume les formules de lien avec les coordonnées de Plücker standard, et donne le schéma de mise à jour locale minimale de la représentation orthonormale ainsi que la matrice Jacobienne $\mathsf{A}^\perp$ exprimant les variations des coordonnées de Plücker par rapport aux paramètres de mise à jour.

6.4.2.5 Interprétation géométrique

Nous donnons une interprétation géométrique des paramètres de la représentation orthonormale et des 4 paramètres de mise à jour. Un des intérêts d'une telle interprétation est de pouvoir incorporer de l'information

connue a priori dans l'estimation d'une droite en n'utilisant que certains des paramètres de mise à jour. Par exemple, étant donnée la direction de la droite, estimer les deux degrés de liberté restant. Lorsque la droite est explicitement paramétrée entre autres par sa direction, l'incorporation de cette information est directe. Lorsque la paramétrisation est basée sur deux projections de la droite, cela devient plus difficile. Lorsque la représentation orthonormale est utilisée, l'interprétation suivante des paramètres de mise à jour fournit une solution simple : les deux degrés de liberté restant peuvent être estimés par les paramètres de mise à jour u_2 et w. Le tableau 6.7 propose différents scénarios d'estimation contrainte et leur réalisation avec la représentation orthonormale. Les notations utilisées ci-dessous sont celles de la figure 2.3 du chapitre 2.

scénario	u_1	u_2	u_3	w
direction fixée		$\times$		$\times$
normale au plan défini par $\boldsymbol{L}$ et l'origine fixée	$\times$			$\times$
distance à l'origine fixée	$\times$	$\times$	$\times$	

TAB. 6.7 – Exemples d'estimations contraintes d'une droite 3D $\boldsymbol{L}$ par mise à jour partielle de la représentation orthonormale. Les croix $\times$ indiquent les paramètres de mise à jour sur lesquelles l'optimisation doit être conduite.

Paramètres de la représentation orthonormale. Soit $(\mathsf{U} = (\mathbf{u}_1 \ \mathbf{u}_2 \ \mathbf{u}_3), \mathsf{W} = \mathsf{R}_{2D}(\alpha)) \in O(3) \times SO(2)$ la représentation orthonormale d'une droite $\boldsymbol{L}$ de coordonnées de Plücker $\mathbf{L}^\mathsf{T} \sim (\ \mathbf{a}^\mathsf{T} \quad \mathbf{b}^\mathsf{T}\ )$. Nous montrons que la matrice U définit l'orientation des trois axes suivants : la direction de $\boldsymbol{L}$, la normale au plan qu'elle définit avec l'origine et la direction du vecteur entre l'origine et le point de $\boldsymbol{L}$ le plus proche. La matrice W représente la distance entre l'origine et $\boldsymbol{L}$. Cette interprétation est illustrée sur la figure 6.2. Pour l'interprétation géométrique des coordonnées de Plücker, nous renvoyons à la figure 2.3 du chapitre 2. D'après l'équation de décomposition manuelle (6.8) et l'unicité de la décomposition QR, si $\|\mathbf{a}\| \neq 0$ et $\|\mathbf{b}\| \neq 0$, les colonnes $\mathbf{u}_1$ et $\mathbf{u}_2$ sont données par $\mathbf{u}_1 = \mathbf{a}/\|\mathbf{a}\|$ et $\mathbf{u}_2 = \mathbf{b}/\|\mathbf{b}\|$. Ces vecteurs correspondent donc respectivement à la direction de la normale à $\boldsymbol{\Pi}$ (le plan formé par $\boldsymbol{L}$ et l'origine $\boldsymbol{O}$) et à la direction de $\boldsymbol{L}$. Comme $\mathsf{U} \in O(3)$, le vecteur $\mathbf{u}_3 = \pm\mathbf{u}_1 \wedge \mathbf{u}_2$ donne la direction de la droite entre $\boldsymbol{O}$ et $\boldsymbol{Q}$, joignant l'origine et le point de $\boldsymbol{L}$ le plus proche de $\boldsymbol{O}$. La matrice $\mathsf{W} = \mathsf{R}_{2D}(\alpha)$ encapsule l'échelle relative entre $\mathbf{a}$ et $\mathbf{b}$ et donc la distance d entre $\boldsymbol{O}$ et $\boldsymbol{L}$. Plus précisément, cette distance est donnée par $d = w_{11}/w_{21} = \tan^{-1}(\alpha)$.

Paramètre de mise à jour w. Ce paramètre met à jour l'angle de la matrice de rotation $\mathsf{W} = \mathsf{R}_{2D}(\alpha)$. Il modifie donc la distance d entre la droite et l'origine : cette dernière passe de $d = \tan^{-1}\alpha$ à $\tan^{-1}(\alpha + w)$. La transformation géométrique correspondant à ce paramètre est un déplacement de la droite parallèlement à elle-même et dans le plan $\boldsymbol{\Pi}$ qu'elle définit avec l'origine. Cette transformation est illustrée sur la figure 6.3 (a).

Paramètre de mise à jour u_1. Le paramètre de mise à jour u_1 correspond à l'application d'une rotation d'axe x à la matrice U. On voit d'après l'équation de recomposition (6.11) que le vecteur $\mathbf{u}_1$ est laissé invariant par cette transformation. Le plan support $\boldsymbol{\Pi}$ de la droite est donc laissé invariant, de même que la distance d, car la matrice W n'est pas modifiée. Le paramètre u_1 affecte donc la direction de la droite. Plus précisément, il correspond à une rotation d'angle u_1 de la droite autour du point $\boldsymbol{Q}$ (le point de $\boldsymbol{L}$ le plus proche de l'origine) et dans le plan $\boldsymbol{\Pi}$. Cette interprétation est locale car le point $\boldsymbol{Q}$ change avec la droite. Soit $\boldsymbol{C}_1$ le cercle centré sur l'origine, contenu par $\boldsymbol{\Pi}$ et tangent à $\boldsymbol{L}$. Sous l'action du paramètre u_1, $\boldsymbol{L}$ tourne autour de $\boldsymbol{C}_1$ avec un angle u_1.

Paramètre de mise à jour u_2. Le paramètre de mise à jour u_2 correspond à l'application d'une rotation d'axe y à la matrice U. On voit d'après l'équation de recomposition (6.11) que le vecteur $\mathbf{u}_2$ est laissé invariant par cette transformation. La direction de la droite est donc laissée invariante, de même que la distance d, car la matrice W n'est pas modifiée. Le paramètre u_2 affecte donc la position de la droite. Soit $\boldsymbol{C}_2$ le cercle centré sur

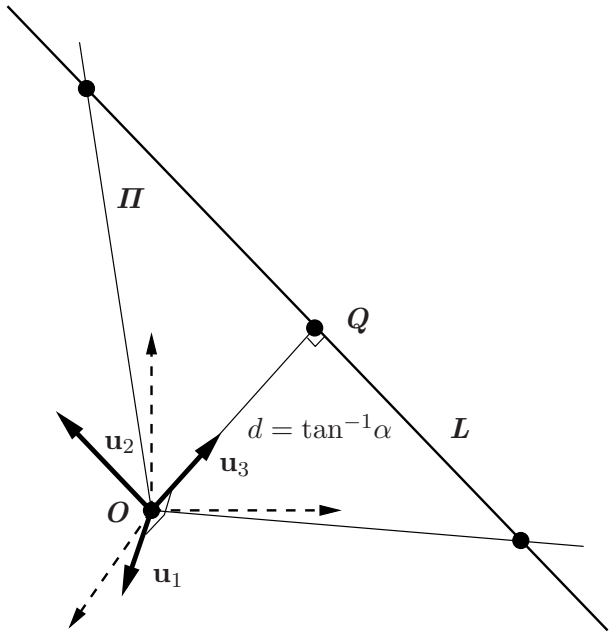


FIG. 6.2 – Interprétation géométrique de la représentation orthonormale $(\mathsf{U} = (\mathbf{u}_1\ \mathbf{u}_2\ \mathbf{u}_3), \mathsf{W} = \mathsf{R}_{2D}(\alpha)) \in O(3) \times SO(2)$ d'une droite 3D $\boldsymbol{L}$. Le vecteur $\mathbf{u}_1$ donne la normale au plan $\boldsymbol{\Pi}$, $\mathbf{u}_2$ la direction de $\boldsymbol{L}$ et $\mathbf{u}_3$ la direction de la droite reliant $\boldsymbol{O}$ et $\boldsymbol{Q}$ (le point de $\boldsymbol{L}$ le plus proche de $\boldsymbol{O}$). La matrice W encapsule la distance d entre $\boldsymbol{O}$ et $\boldsymbol{L}$: $d = \tan^{-1}\alpha$.

l'origine, dont le plan support est orthogonal à $\boldsymbol{\Pi}$ et parallèle à $\boldsymbol{L}$, et de rayon d. La droite $\boldsymbol{L}$ est orthogonale à une tangente au cercle $\boldsymbol{C}_2$. Sous l'action du paramètre u_2, $\boldsymbol{L}$ tourne autour de $\boldsymbol{C}_2$ avec un angle u_2.

Paramètre de mise à jour u_3. Le paramètre de mise à jour u_3 correspond à l'application d'une rotation d'axe z à la matrice U. On voit d'après l'équation de recomposition (6.11) qu'aucun des vecteurs $\mathbf{u}_1$ et $\mathbf{u}_2$ n'est laissé invariant par cette transformation. Les vecteurs obtenus après transformation sont des combinaisons linéaires des vecteurs $\mathbf{u}_1$ et $\mathbf{u}_2$ originaux et se trouvent donc dans le plan formé par les vecteurs $\mathbf{u}_1$ et $\mathbf{u}_2$ originaux. Le paramètre u_3 réalise donc une rotation d'angle u_3 de la droite autour de l'axe défini par les points $\boldsymbol{O}$ et $\boldsymbol{Q}$.

6.4.2.6 Droites à l'infini ou contenant l'origine

Les droites 3D sur le plan à l'infini d'une part, et les droites 3D contenant l'origine du repère monde d'autre part n'ont pas une représentation orthonormale unique. Plus précisément, leur représentation orthonormale a un degré d'ambiguïté.

Caractérisation des ambiguïtés. Pour caractériser analytiquement ces ambiguïtés, considérons l'équation de recomposition (6.11) :

$$\mathbf{L}(\mathsf{U},\mathsf{W})^\mathsf{T} \quad = \quad (\ w_{11}\mathbf{u}_1^\mathsf{T} \qquad w_{21}\mathbf{u}_2^\mathsf{T}\ ).$$

Lorsque $w_{11} = 0$, c'est à dire lorsque la droite contient l'origine, alors les coordonnées de Plücker s'écrivent $(\ \mathbf{0}^\mathsf{T} \quad w_{21}\mathbf{u}_2^\mathsf{T}\ )^\mathsf{T}$. Elles ne dépendent plus de $\mathbf{u}_1$ mais seulement de $\mathbf{u}_2$. La matrice U peut donc être transformée librement, sans changer la droite représentée par toute matrice de rotation laissant $\mathbf{u}_2$ invariant. Ce sont toutes les rotations autour de l'axe des y, notées $\mathsf{R}_y(\alpha)$. Cette ambiguïté est résumée par l'équation suivante :

$$(\boldsymbol{O} \in \boldsymbol{L}) \Leftrightarrow (w_{11} = 0) \quad \Rightarrow \quad (\forall \alpha \in \mathbb{R},\ \mathbf{L}(\mathsf{U},\mathsf{W}) = \mathbf{L}(\mathsf{U}\mathsf{R}_y(\alpha),\mathsf{W}))\,.$$

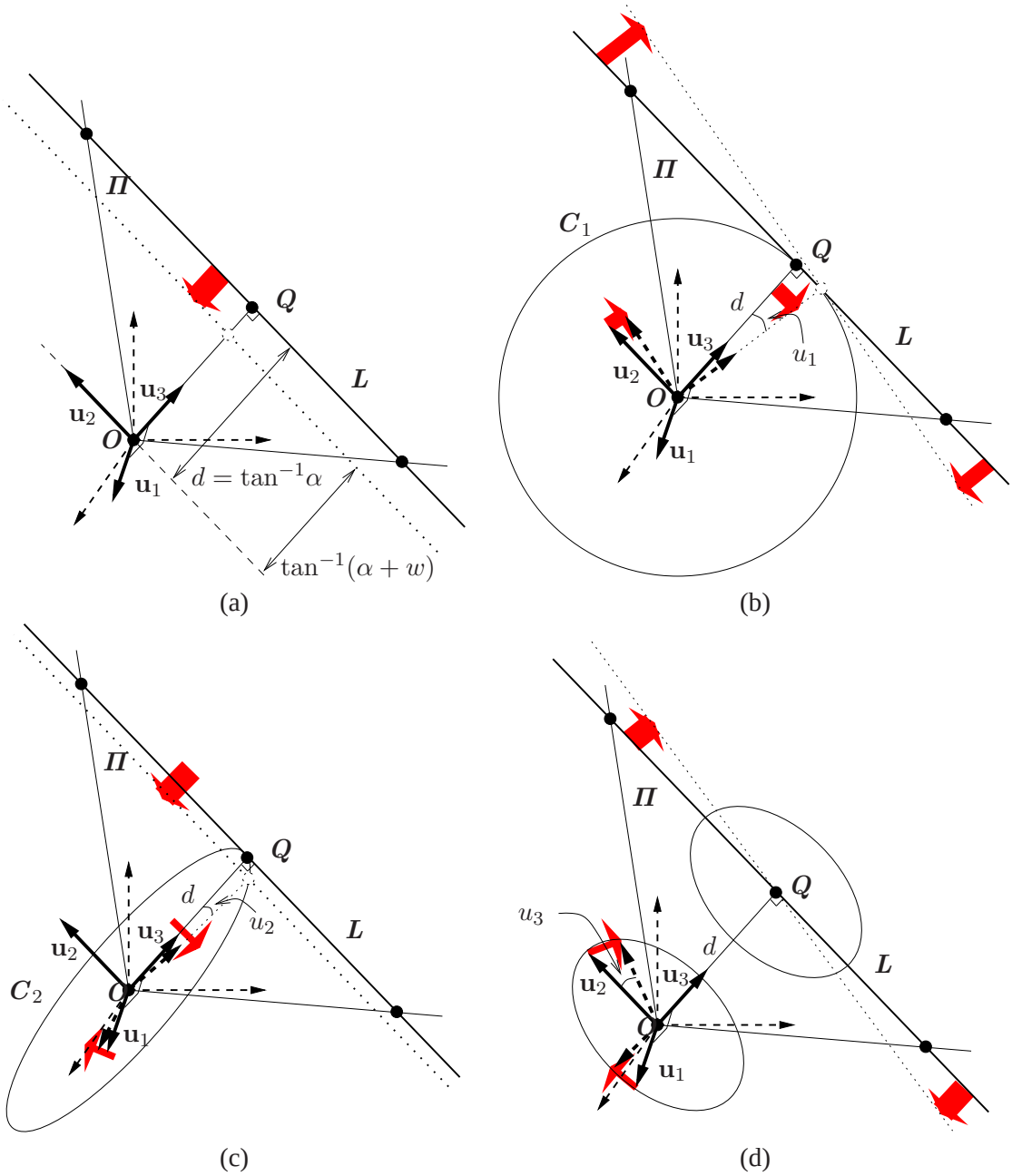


FIG. 6.3 – Interprétation géométrique de la mise à jour de la représentation orthonormale $(\mathsf{U} = (\mathbf{u}_1\ \mathbf{u}_2\ \mathbf{u}_3), \mathsf{W} = \mathsf{R}_{2D}(\alpha))$ d'une droite 3D $\boldsymbol{L}$ selon le schéma de mise à jour (6.13) de paramètres $\mathbf{x}^\mathsf{T} = (u_1\ u_2\ u_3\ w)$. Les flèches rouges indiquent le mouvement subi par la droite. (a) : le paramètre w modifie la distance entre $\boldsymbol{L}$ et $\boldsymbol{O}$ ($\boldsymbol{L}$ est translatée dans le plan $\boldsymbol{\Pi}$ parallèlement à elle-même). (b) : le paramètre u_1 provoque une rotation d'angle u_1 de $\boldsymbol{L}$ autour du cercle $\boldsymbol{C}_1$ de centre $\boldsymbol{O}$, de rayon d et contenu dans le plan $\boldsymbol{\Pi}$. (c) : le paramètre u_2 provoque une rotation d'angle u_2 de $\boldsymbol{L}$ autour du cercle $\boldsymbol{C}_2$ de centre $\boldsymbol{O}$, de rayon d et contenu dans le plan défini par $\mathbf{u}_1$ et $\mathbf{u}_3$. (d) : le paramètre u_3 provoque une rotation d'angle u_3 de $\boldsymbol{L}$ autour de l'axe défini par les points $\boldsymbol{O}$ et $\boldsymbol{Q}$.

Lorsque $w_{21} = 0$, c'est à dire lorsque la droite est à l'infini, les coordonnées de Plücker s'écrivent $\begin{pmatrix} w_{11}\mathbf{u}_1^\mathsf{T} & \mathbf{0}^\mathsf{T} \end{pmatrix}^\mathsf{T}$ et ne dépendent plus de $\mathbf{u}_2$. La matrice U peut donc être multipliée par toute matrice de rotation autour de l'axe des x, laissant invariant $\mathbf{u}_1$:

$$(\boldsymbol{L} \in \boldsymbol{\Pi}_\infty) \Leftrightarrow (w_{21} = 0) \quad \Rightarrow \quad (\forall \alpha \in \mathbb{R},\ \mathbf{L}(\mathsf{U}, \mathsf{W}) = \mathbf{L}(\mathsf{U}\mathsf{R}_x(\alpha), \mathsf{W}))\,.$$

Pour résumer, la représentation orthonormale est ambiguë lorsque $u_{11}w_{12} = 0$.

Conséquences sur l'optimisation non-linéaire. Une des conséquences de ces ambiguïtés est le fait qu'elles rendent la matrice Jacobienne $\mathsf{A}^\perp$, dont l'expression est donnée par l'équation (6.16), singulière ou mal conditionnée. Ceci ne pose pas de problème lorsqu'une technique d'optimisation telle Levenberg-Marquardt, basée sur une régularisation des équations normales, est utilisée. Nous avons montré expérimentalement dans le cas de la représentation orthonormale de la matrice fondamentale, voir chapitre 4, §4.4.8 que de telles ambiguïtés dans la représentation ne posent pas de problèmes pratiques pour l'optimisation non-linéaire.

6.5 Résultats expérimentaux

Nous comparons nos algorithmes avec d'autres à l'aide de données simulées puis réelles. Nous détaillons ci-dessous les méthodes comparées, l'erreur mesurée, la reconstruction initiale et la paramétrisation des caméras pour l'ajustement de faisceaux.

Méthodes comparées. Nous comparons les méthodes suivantes. Le premier groupe de méthodes n'agit que sur la structure. Ce sont des méthodes de triangulation, c'est à dire de reconstruction individuelle de droites, dont le résultat dépend du mouvement des caméras :

- ▷ LIN_HARTLEY, voir §6.3.1, est la méthode de triangulation proposée dans [92, 93, §11.6] basée sur les représentations duales (deux points ou deux plans) ;
- ▷ LIN, voir le tableau 6.3, est la méthode de reconstruction de coordonnées de Plücker, avec post-correction de Plücker ;
- ▷ QLIN1, voir le tableau 6.4, est la correction quasi-linéaire du biais introduit par la distance algébrique dans LIN, alternée avec une correction de Plücker ;
- ▷ QLIN2, voir le tableau 6.5, complète QLIN1 en incorporant la contrainte de Plücker au premier ordre à chaque itération ;
- ▷ NLIN, voir §6.3.8, est la minimisation non-linéaire par l'algorithme de Levenberg-Marquardt de l'erreur de reprojection, basée sur la représentation orthonormale de la droite décrite dans le tableau 6.6.

Le deuxième groupe de méthodes agit sur la structure et le mouvement des caméras. Ce sont les méthodes d'ajustement de faisceaux :

- ▷ MLE_HARTLEY est la méthode proposée dans [92] d'estimation du tenseur trifocal et des droites paramétrées par deux de leurs images, dont une extension au cas de vues multiples est proposée en §6.4.1 ;
- ▷ MLE est la minimisation non-linéaire par l'algorithme de Levenberg-Marquardt de l'erreur de reprojection, basée sur la représentation orthonormale décrite dans le tableau 6.6.

Erreur mesurée. Nous mesurons la qualité d'une estimation en utilisant l'*erreur d'estimation*, comme décrit dans [93, §4]. Ceci fournit aussi la borne inférieure théorique. L'erreur d'estimation est équivalente à l'erreur de reprojection et se calcule directement par la formule (6.7) avec les valeurs du mouvement et de la structure obtenues à la convergence des algorithmes.

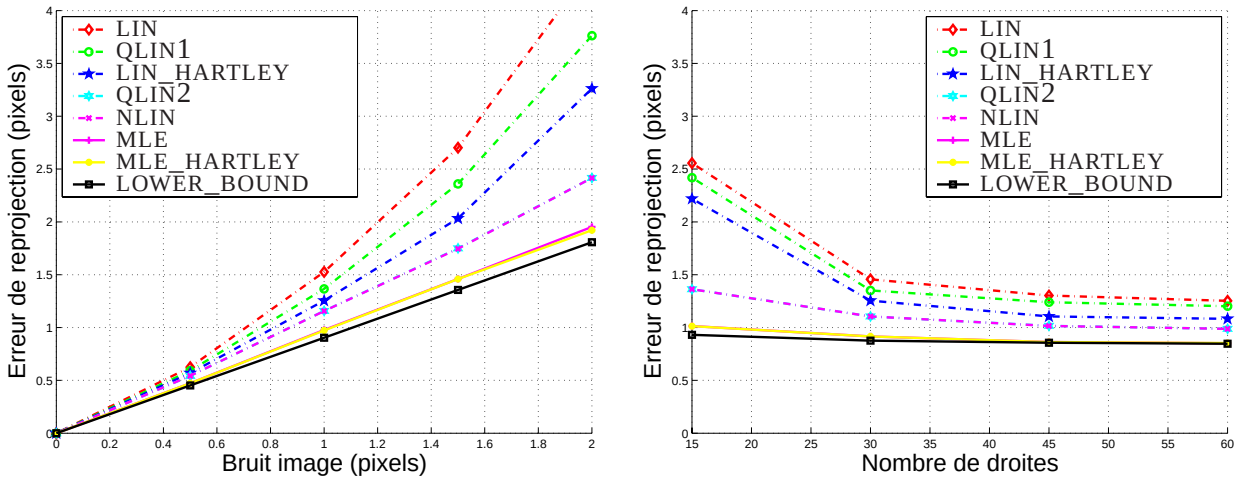
Reconstruction initiale des caméras. La reconstruction initiale des caméras est effectuée par la méthode de Triggs [216] basée sur les contraintes de fermeture. Ces contraintes de fermeture sont formulées en termes des tenseurs trifocaux entre triplets d'images, dont le calcul est effectué par l'algorithme de Hartley [92].

Paramétrisation des caméras pour l'ajustement de faisceaux. Nous paramétrisons le mouvement des caméras pour l'estimation non-linéaire comme indiqué au chapitre 3, §3.5.5. Brièvement, la paramétrisation consiste à prendre pour paramètres les coefficients de $n-1$ matrices de projections.

6.5.1 Données simulées

Scène simulée. Nous simulons un jeu de caméras observant un ensemble de droites 3D, choisies au hasard dans une sphère d'un rayon de 1 mètre, par tirage aléatoire de leurs extrémités. Les caméras sont bien reparties autour de cette sphère. Nous fixons la distance focale des caméras à 1000 (en pixels). Notons que cette information n'est pas utilisée dans le reste des expérimentaux : les reconstructions obtenues sont projectives. Les extrémités de toutes les droites sont projetés dans toutes les vues et leurs projections sont bruitées avec une distribution Gaussienne. La configuration standard consiste en 3 caméras situées à une distance de 10 mètres des droites, et une distance entre caméras consécutives ("baseline" en Anglais) de 1 mètre. Le nombre de droites simulées est 30 et le niveau de bruit est 1 pixel. Chaque paramètre de cette scène est varié indépendamment pour pouvoir comparer les différentes méthodes dans différentes situations. Les résultats sont des moyennes sur 100 essais.

Résultats. Les figures 6.4 et 6.5 montrent les résultats obtenus lorsque différents paramètres d'observation sont modifiés. Nous commentons ces résultats de manière globale puis spécifiquement. Nous observons que



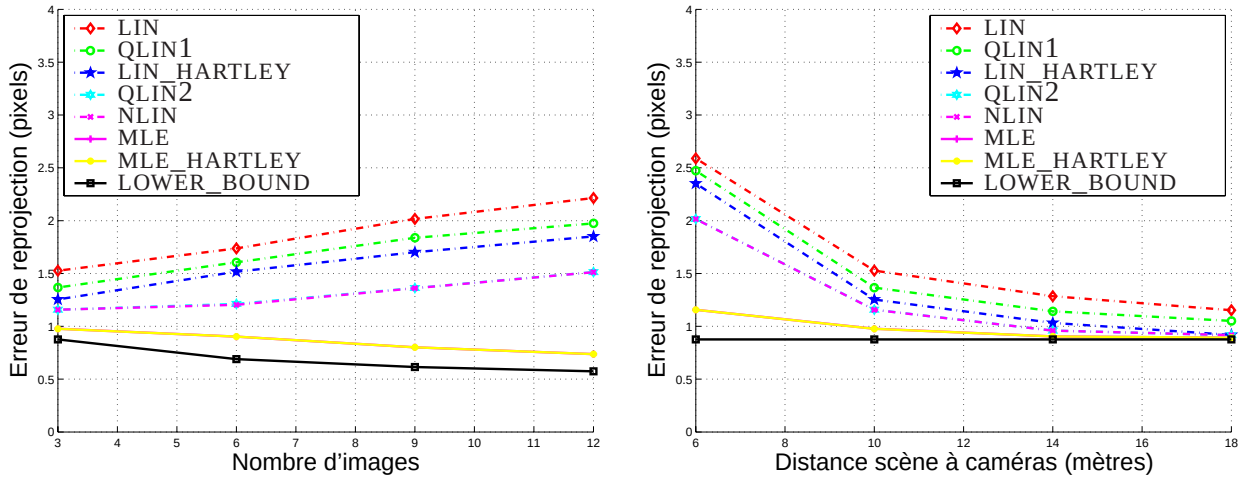
(a) (b)

FIG. 6.4 – Erreur de reprojection pour différents niveaux de bruit (a) et différents nombres de droites (b).

l'ordre des méthodes est toujours le même. On peut repartir les méthodes en trois groupes.

Les méthodes LIN et QLIN1 donnent des résultats très proches. Elles sont suivies par LIN_HARTLEY. Ceci s'explique comme suit. Les méthodes LIN et QLIN1 donnent un résultat biaisé par la correction de Plücker. De plus, la méthode LIN minimise une erreur algébrique. La méthode LIN_HARTLEY n'a pas le biais introduit par la correction de Plücker mais est aussi biaisée pour deux raisons : d'une part, l'erreur minimisée est algébrique, et d'autre part, la forme de cette erreur, voir §6.3.1 est "duale" à celle de l'erreur de reprojection (elle est exprimée entre points reprojetés et droites observées, contrairement à l'utilisation de droites reprojetées et points mesurés), contrairement à l'erreur considérée par LIN, équation (6.4).

Ensuite, les méthodes QLIN2 et NLIN donnent des résultats intermédiaires entre les méthodes du groupe précédent et l'ajustement de faisceaux. Les courbes pour ces deux méthodes sont confondues. Ceci est logique car elles minimisent toutes deux l'erreur de reprojection individuellement pour chaque droite et prennent toutes deux en compte la contrainte de Plücker, NLIN2 via sa linéarisation, et NLIN par une paramétrisation adaptée, la représentation orthonormale des droites 3D.



(a) (b)

FIG. 6.5 – Erreur de reprojection pour différents nombres d'images (a) et distances scène à caméras (b).

Finalement, les méthodes MLE et MLE_HARTLEY d'ajustement de faisceaux, impliquant les droites et les caméras, donnent des résultats souvent identiques. En effet, seule la paramétrisation des droites change entre ces deux méthodes, MLE utilisant la représentation orthonormale et MLE_HARTLEY utilisant deux droites image. Nous n'avons pas rencontré les singularités de la représentation par deux droites image au cours de ces expérimentations.

Plus spécifiquement, nous observons sur la figure 6.4 (a) qu'avec plus d'un pixel de bruit, les méthodes du premier groupe donnent des résultats se dégradant rapidement, alors que les autres méthodes se dégradent plus harmonieusement à mesure que le bruit augmente. Les méthodes d'ajustement de faisceaux donnent des résultats très proches du minimum théorique. Des remarques similaires peuvent être faites lorsque le nombre de droites est varié. Cependant, on observe sur la figure 6.4 (b) que lorsque plus de 30 droites sont considérées, les méthodes du premier groupe donnent des résultats raisonnables. Ceci s'explique de la manière suivante. Ces méthodes procèdent indépendamment pour chaque droite et ne sont sensibles aux changements du nombre de droites que parce qu'elle dépendent d'une estimation du mouvement des caméras, dont la qualité dépend elle-même du nombre de droites. Lorsque plus de 45 droites sont présentes, les méthodes d'ajustement de faisceaux donnent des résultats confondus avec le minimum théorique.

Lorsque le nombre de caméras augmente, figure 6.5 (a), les algorithmes ne modifiant pas le mouvement voient leurs erreurs augmenter, alors que les algorithmes d'ajustement de faisceaux suivent une courbe descendante. Ceci est dû au fait que lorsque le nombre d'images augmente, le calcul initial du mouvement se dégrade. Ceci est une caractéristique de l'algorithme d'initialisation du mouvement. Il est donc normal que les méthodes ne modifiant pas cette estimation initiale donnent des résultats allant en se dégradant. Lorsque la distance entre les droites et les caméras augmente, figure 6.5 (b), l'erreur diminue pour toutes les méthodes. Ceci s'explique par le fait que le nuage de droites observé devient de plus en plus petit dans les images, ce qui réduit l'erreur absolue, mais ne traduit pas nécessairement une amélioration de la qualité de l'estimation.

6.5.2 Données réelles

Nous avons testé et comparé nos algorithmes sur plusieurs séquences d'images. Notons que la partie expérimentale du chapitre 8, utilise les modèles 3D dont l'obtention est décrite ci-dessous. En particulier, il y est réalisé l'alignement des séquences des livres I et des livres II et des séquences des boîtes I et des boîtes II. On y montre un autre exemple de reconstruction de droites, la séquence de la pièce de bateau.

6.5.2.1 Séquence des livres I

Cette séquence est composée des 5 images montrées sur la figure 6.6. Nous donnons manuellement 45 cor-

FIG. 6.6 – Les 5 images de la séquence des livres I, avec les 45 correspondances de droites utilisées.

respondances de droites entre ces images. Certaines droites ne sont visibles que dans deux vues et ne fournissent pas de contraintes sur le mouvement. Pour ces droites, il existe une solution de reconstruction exacte. Nous estimons le mouvement projectif des caméras à l'aide des contraintes de fermeture comme décrit en §6.1. Nous utilisons la méthode proposée par Pollefeys et al. [160] basée sur la quadrique absolue pour obtenir une conversion métrique de la reconstruction. Nous avons donc une estimation des caméras en espace métrique. Nous utilisons nos algorithmes pour calculer la structure à partir de ces caméras. Les erreurs d'estimation obtenues pour les différents algorithmes sont données dans le tableau 6.8. Les algorithmes d'ajustement de faisceaux MLE et MLE_HARTLEY sont initialisés avec le résultat de reconstruction de QLIN2. Pour chaque algorithme,

algorithme	LIN	QLIN1	LIN_HARTLEY	QLIN2	NLIN	MLE	MLE_HARTLEY
erreur de reprojection (pixels)	2,30	2,27	1,85	1,43	1,43	0,94	0,94

TAB. 6.8 – Erreurs de reprojection (pixels) obtenues lors de la reconstruction des droites de la séquence des livres I avec différents algorithmes.

nous reconstruisons les extrémités des droites correspondant à la première image (en haut à gauche de la figure 6.6). Les meilleurs points, au sens du maximum de vraisemblance, sont donnés en projetant orthogonalement leurs images sur l'image de la droite correspondante. La figure 6.7 montre quelques vues du modèle 3D ainsi obtenu. La figure 6.8 montre les segments reconstruits puis reprojetés dans une image originale. Ceci permet

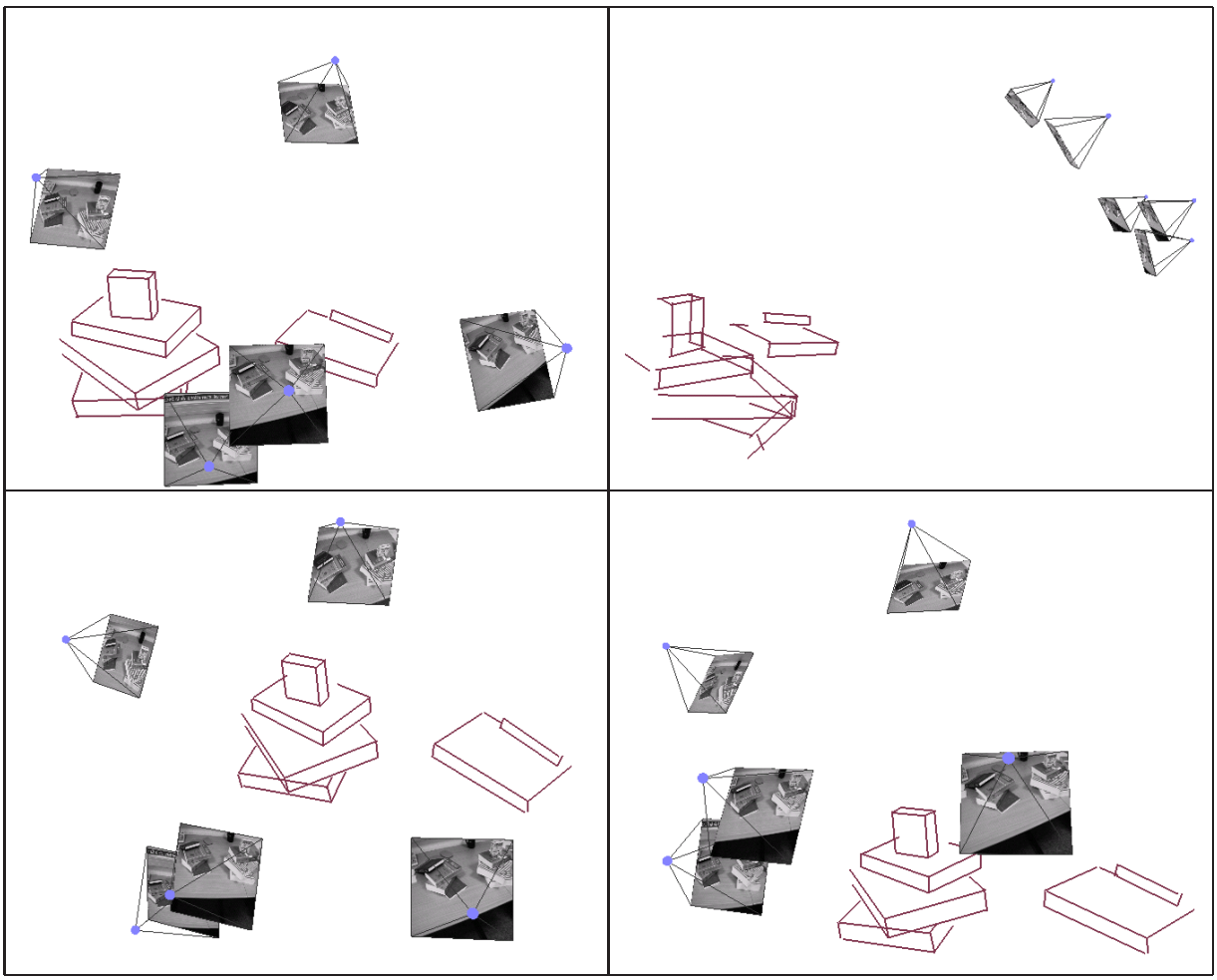


FIG. 6.7 – Quelques vues de la reconstruction métrique obtenue par ajustement de faisceaux pour la séquence des livres I. Les extrémités des segments sont données par la première image de la séquence.

de comparer visuellement les points mesurés et les droites prédites par le modèle. Nous notons des différences importantes entre les différents algorithmes. En particulier, les méthodes QLIN2 et NLIN améliorent grandement les résultats des méthodes LIN, QLIN1 et LIN_HARTLEY. Les droites originales et celles prédites par l'ajustement de faisceaux sont confondues.

6.5.2.2 Séquence des livres II

Cette séquence est composée de 4 images, montrées sur la figure 6.9, de la même scène que celles de la séquence des livres I, mais de points de vue très éloignés. En suivant la même démarche expérimentale que

LIN & QLIN1 LIN_HARTLEY

QLIN2 & NLIN MLE & MLE_HARTLEY

FIG. 6.8 – Zoom sur la reprojection des droites dans les images originales pour différentes méthodes, pour la séquence des livres I. Les droites originales sont en blanc et les droites reprojetées sont en noir.

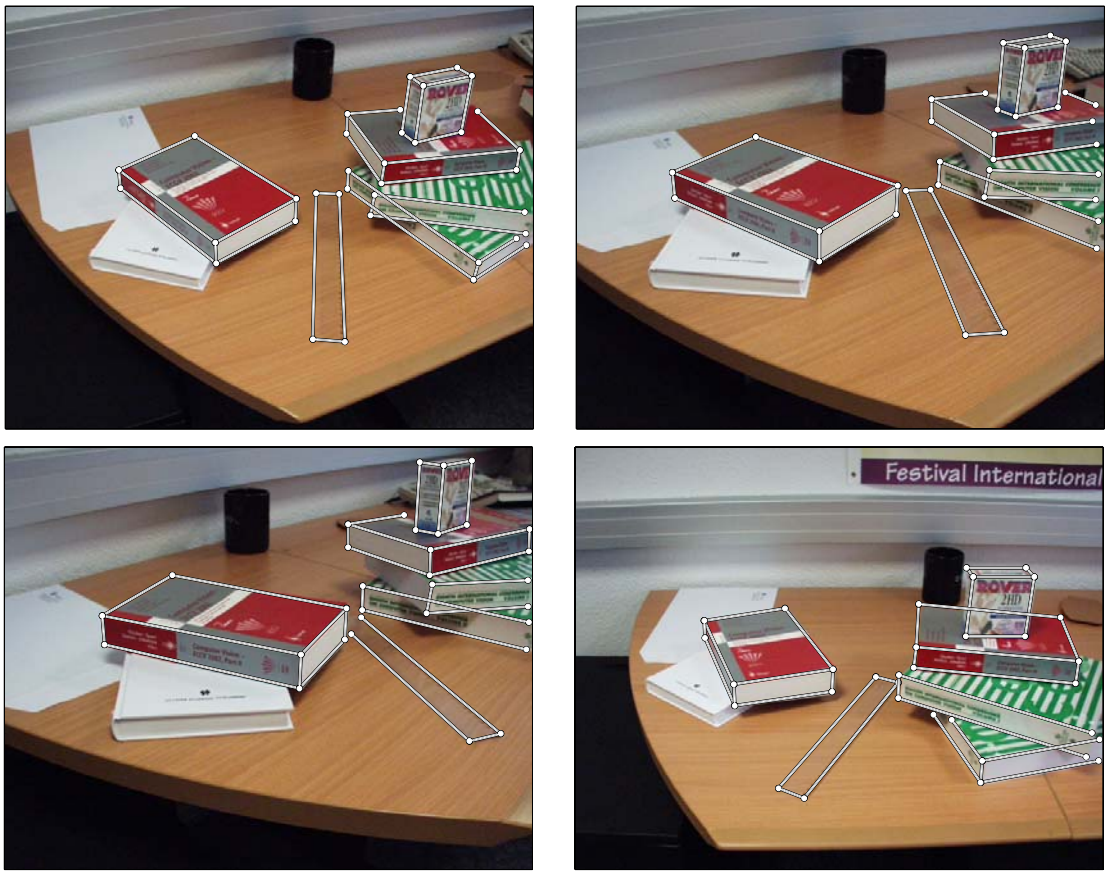


FIG. 6.9 – Les 4 images de la séquence des livres II, avec les 40 correspondances de droites utilisées.

dans la section précédente pour la séquence des livres I, nous obtenons une reconstruction métrique, dont des captures sont visibles sur la figure 6.10. Les erreurs de reprojection obtenues pour les différents algorithmes

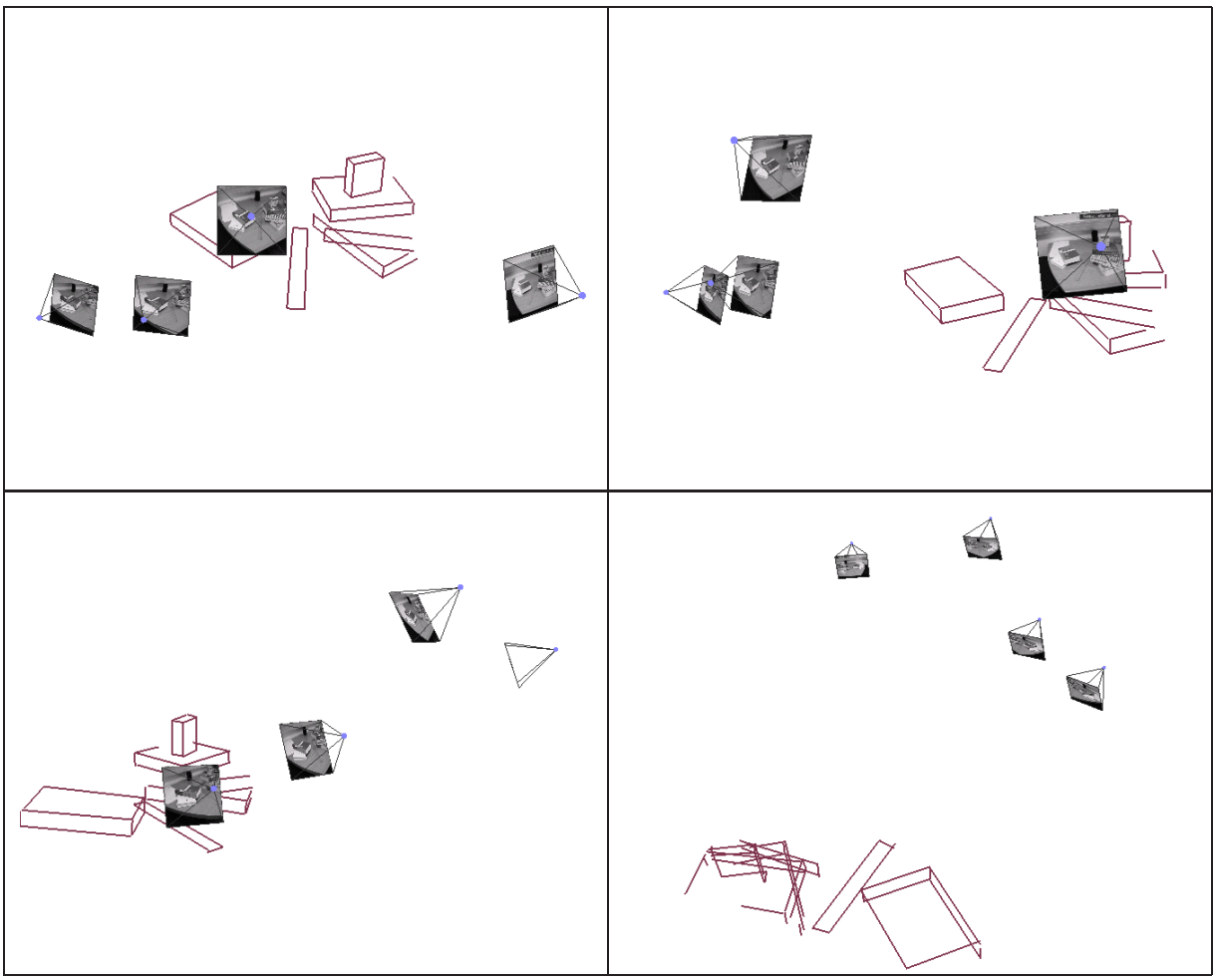


FIG. 6.10 – Quelques vues de la reconstruction métrique obtenue par ajustement de faisceaux pour la séquence des livres II. Les extrémités des segments sont données par la première image de la séquence.

sont données dans le tableau 6.9.

algorithme	LIN	QLIN1	LIN_HARTLEY	QLIN2	NLIN	MLE	MLE_HARTLEY
erreur de reprojection (pixels)	0,79	0,76	0,71	0,63	0,63	0,51	0,51

TAB. 6.9 – Erreurs de reprojection (pixels) obtenues lors de la reconstruction des droites de la séquence des livres II avec différents algorithmes.

6.5.2.3 Séquence des boîtes I

Cette séquence est composée de 8 images, montrées sur la figure 6.11. En suivant la même démarche expérimentale que dans les sections précédentes pour les séquences des livres, nous obtenons une reconstruction métrique, dont des captures sont visibles sur la figure 6.12. Les erreurs de reprojection obtenues pour les différents algorithmes sont données dans le tableau 6.10.

algorithme	LIN	QLIN1	LIN_HARTLEY	QLIN2	NLIN	MLE	MLE_HARTLEY
erreur de reprojection (pixels)	1,34	1,29	1,03	1,04	1,03	0,82	0,82

TAB. 6.10 – Erreurs de reprojection (pixels) obtenues lors de la reconstruction des droites de la séquence des boîtes I avec différents algorithmes.

6.5.2.4 Séquence des boîtes II

Cette séquence est composée de 6 images, montrées sur la figure 6.13. La scène observée est la même que pour la séquence des boîtes I, mais de points de vue différents. En suivant la même démarche expérimentale que dans les sections précédentes, nous obtenons une reconstruction métrique, dont des captures sont visibles sur la figure 6.14. Les erreurs de reprojection obtenues pour les différents algorithmes sont données dans le tableau 6.11.

algorithme	LIN	QLIN1	LIN_HARTLEY	QLIN2	NLIN	MLE	MLE_HARTLEY
erreur de reprojection (pixels)	1,36	1,23	1,11	0,99	0,99	0,87	0,87

TAB. 6.11 – Erreurs de reprojection (pixels) obtenues lors de la reconstruction des droites de la séquence des boîtes II avec différents algorithmes.

6.6 Conclusions et perspectives

Conclusions. Nous avons abordé le problème de la reconstruction de droites à partir de correspondances multi-vues et plus particulièrement la reconstruction initiale par triangulation et l'ajustement de faisceaux final. La résolution de ces problèmes soulève deux interrogations principales : la représentation d'une droite 3D et l'expression de sa projection perspective. Nos choix et propositions sont basés sur une étude des différentes représentations.

Concernant le problème de la triangulation, c'est à dire la reconstruction de droites étant donnée une estimation du mouvement de caméra, le but est de minimiser l'erreur de reprojection, ce qui nécessite une représentation des droites 3D en terme de laquelle les coordonnées des droites reprojetées s'expriment de manière linéaire. Les coordonnées de Plücker sont une des rares représentations satisfaisant cette propriété. Nous avons développé plusieurs algorithmes basés sur cette représentation. La minimisation de l'erreur de reprojection est basée sur un schéma d'optimisation quasi-linéaire (moindres carrés repondérés itérativement) couplé avec une linéarisation de la contrainte de Plücker.

Nous avons ensuite abordé le problème de l'ajustement de faisceaux, impliquant caméras et droites. Les représentations existantes des droites 3D ne sont pas bien adaptées à l'optimisation non-linéaire dans le sens où elles impliquent des libertés de jauge, nécessitent des contraintes internes et un nombre redondant de paramètres pour l'optimisation. Nous avons proposé la représentation orthonormale des droites 3D, qui permet l'optimisation non-linéaire avec le nombre minimum de 4 paramètres, sans contrainte interne ou liberté de jauge. Elle permet en outre de laisser invariantes certaines parties de la droite, par exemple sa direction.

La comparaison de nos algorithmes avec d'autres permet de dégager les conclusions suivantes. Concernant la triangulation, la méthode linéaire donnant les meilleurs résultats est celle de Hartley. Cependant, nous observons que toutes les méthodes linéaires supportent mal la dégradation des conditions d'observation. Les résultats obtenus par la méthode quasi-linéaire proposée sont bien meilleurs, et équivalents à une minimisation directe de l'erreur de reprojection par l'algorithme de Levenberg-Marquardt.

Concernant l'ajustement de faisceaux, notre méthode basée sur la représentation orthonormale des droites 3D et la méthode de Hartley basée sur une représentation par deux droites image donnent les mêmes résultats. Cependant, notre représentation orthonormale est complète par opposition à celle de Hartley qui a des singularités importantes.

Perspectives. Au cours de ce travail, nous avons noté qu'une autre représentation orthonormale des droites 3D pouvait être définie, en partant de la matrice de Plücker L. Cette matrice est anti-symétrique et de rang 2 et il est facile de montrer, par exemple à l'aide d'un logiciel de calcul symbolique tel MAPLE, que ses 2 valeurs singulières non nulles sont égales. La décomposition en valeurs singulières de L s'écrit donc :

$$\mathsf{L}_{(4\times4)} \quad \sim \quad \mathsf{U}_{(4\times4)}\, \mathrm{diag}(1,1,0,0)\, \mathsf{V}^{\mathsf{T}}_{(4\times4)}.$$

Chaque matrice orthonormale U et V a 6 degrés de liberté, correspondant aux rotations sur les 6 axes élémentaires, voir la section 2.7.2 du chapitre 2. Il est facile de montrer que seules 4 rotations élémentaires sont indépendantes, ce qui fournit un schéma de mise à jour minimale et locale de cette représentation orthonormale.

FIG. 6.11 – Les 8 images de la séquence des boîtes I, avec les 40 correspondances de droites utilisées.

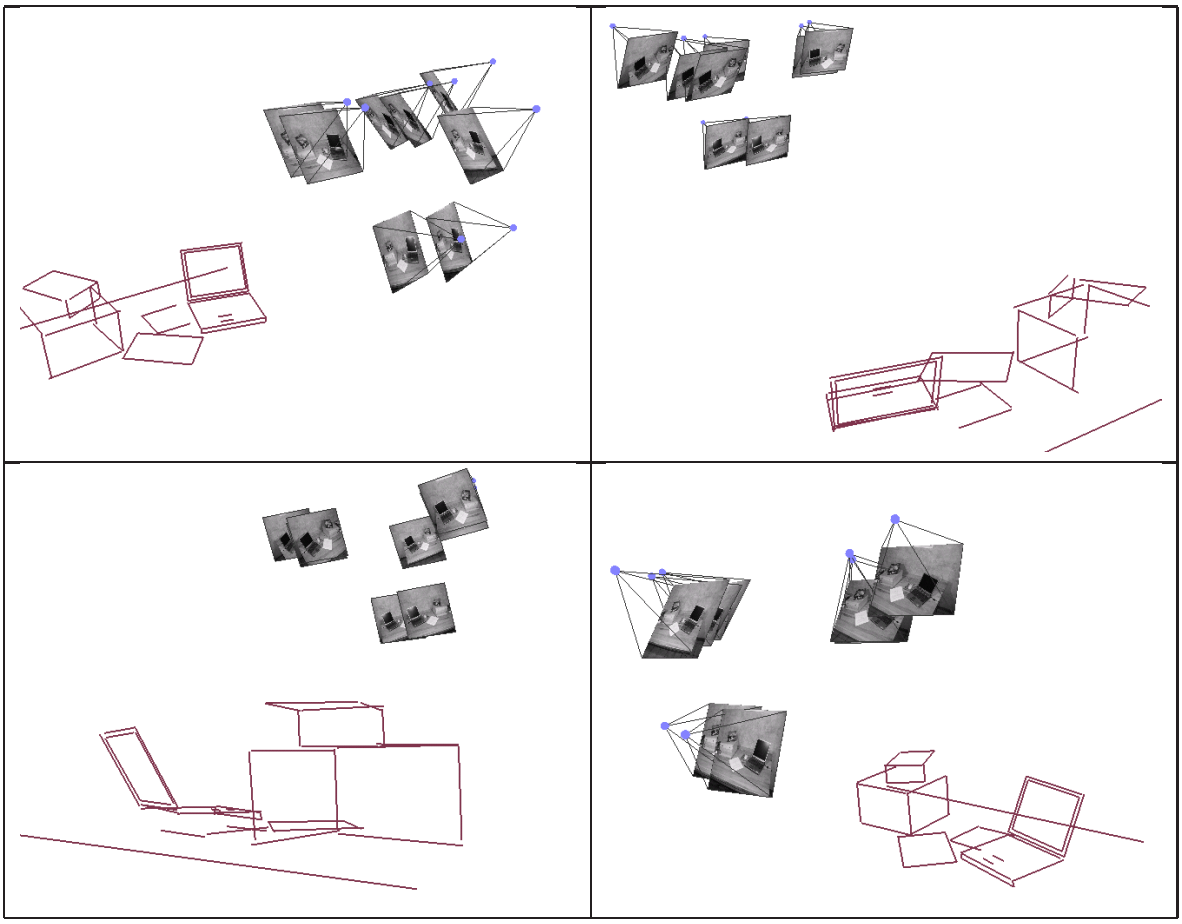


FIG. 6.12 – Quelques vues de la reconstruction métrique obtenue par ajustement de faisceaux pour la séquence des boîtes I. Les extrémités des segments sont données par la première image de la séquence.

FIG. 6.13 – Les 6 images de la séquence des boîtes II, avec les 40 correspondances de droites utilisées.

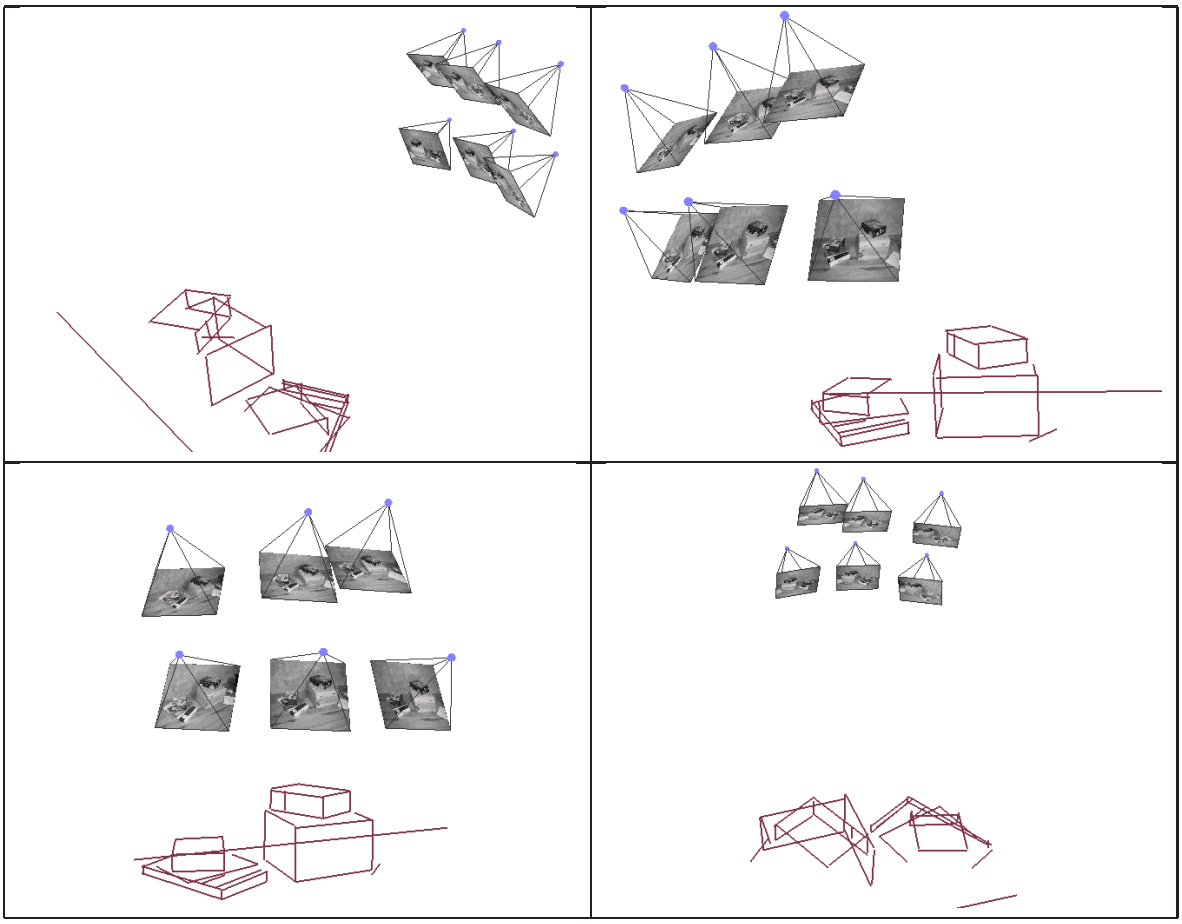


FIG. 6.14 – Quelques vues de la reconstruction métrique obtenue par ajustement de faisceaux pour la séquence des boîtes II. Les extrémités des segments sont données par la première image de la séquence.

CHAPITRE

7

LES MATRICES DE MOUVEMENT POUR DROITES 3D

Ce chapitre est consacré à l'étude de l'application d'un mouvement à une droite en 3D. Nous avons vu au chapitre 6 que la représentation des droites 3D était un sujet délicat. Nous nous concentrons sur les coordonnées de Plücker. Nos contributions sont la formulation et l'étude de matrices de taille (6×6) permettant l'application d'un mouvement 3D directement aux coordonnées de Plücker. Nous appelons ces matrices les *matrices de mouvement pour droites 3D*. Elles dépendent directement des matrices de mouvement usuelles (4×4). Nous caractérisons leurs propriétés algébriques et géométriques et proposons plusieurs algorithmes de manipulation, en particulier pour l'extraction d'un mouvement à partir d'une matrice de mouvement pour droites 3D.

Ce travail a été effectué en collaboration avec Peter Sturm et a été publié dans [24, 20, 23].

7.1 Introduction

La représentation usuelle du mouvement 3D est une matrice (4×4) définie à un facteur près. Cette représentation est utilisable pour représenter, entre autres, des mouvements de type homographique, affine, similitude et rigide. L'action du mouvement sur un point 3D représenté en coordonnées homogènes est directe. Soit G une matrice de mouvement usuelle (4×4) et $\mathbf{Q}$ les coordonnées homogènes d'un point 3D, alors les coordonnées du point transféré sont données par $\mathsf{G}\mathbf{Q}$. De même, cette représentation agit directement sur les coordonnées homogènes représentant des plans. Soit $\mathbf{\Pi}$ les coordonnées homogènes d'un plan, les coordonnées du plan transféré sont données par $\mathsf{G}^{-\mathsf{T}}\mathbf{\Pi}$.

Dans ce chapitre, nous abordons le problème de l'application d'un mouvement 3D à une droite 3D, c'est à dire du transfert d'une droite 3D d'une base de l'espace à une autre, d'un point de vue algébrique. Les applications directes d'une telle étude sont variées, et comptent en particulier l'alignement de reconstructions de droites et le calibrage en ligne des caméras. Ces applications sont étudiées au chapitre suivant.

Le transfert d'une droite d'une base de l'espace à une autre, étant donnée la matrice de mouvement usuelle reliant ces deux bases, peut prendre différentes formes, selon la représentation choisie pour la droite. On se reportera à l'étude des représentations des droites 3D du chapitre 6, §6.2, pour une présentation plus complète des différentes représentations. Par exemple, lorsque la droite est représentée par deux points ou deux plans, le transfert est effectué en transférant ces deux points ou ces deux plans. Lorsque la droite est représentée par une matrice de Plücker, voir la section 2.12 du chapitre 2, le transfert est effectué via l'équation (2.57). Ce transfert est bilinéaire en les coefficients de la matrice de mouvement usuelle.

Nous nous concentrons sur une représentation des droites en coordonnées de Plücker. Ces coordonnées sont encapsulées dans un vecteur-6 soumis à une contrainte bilinéaire, la contrainte de Plücker. Elles sont présentées en détails en §2.12, chapitre 2 et ont été utilisées pour la reconstruction de droites au chapitre précédent. Ces coordonnées ont l'avantage de ne pas dépendre des points ou plans utilisés pour définir la droite. Les méthodes actuelles ne permettent pas le transfert directe d'une droite exprimée en coordonnées de Plücker. Il faut alors passer par une représentation intermédiaire pour réaliser le transfert, par exemple retrouver deux points ou deux plans sur la droite, les transférer puis former les coordonnées de Plücker associées, ou former et transférer la matrice de Plücker puis en extraire les coordonnées de Plücker.

Une exception au constat ci-dessus est le cas de l'application d'un mouvement rigide à une droite exprimée en coordonnées de Plücker. Navab et Faugeras [152] montrent qu'une matrice (6×6) permet de réaliser directement le transfert, comme suit. Considérons une droite 3D $\boldsymbol{L}$ avec les coordonnées de Plücker $\mathbf{L}^{\mathsf{T}} \sim (\ \mathbf{a}^{\mathsf{T}} \quad \mathbf{b}^{\mathsf{T}}\ )$ et $\mathbf{L}'^{\mathsf{T}}$dans deux bases reliées par une transformation rigide représentée par une matrice de rotation R et un vecteur de translation $\mathbf{t}$. Nous nous basons sur l'interprétation géométrique des coordonnées de Plücker afin d'appliquer la transformation rigide directement à ces coordonnées. La partie inférieure, le vecteur $\mathbf{b}$, représente la direction de la droite, en d'autres termes, son point à l'infini. Lors de l'application d'un mouvement rigide, il n'est donc pas affecté par la translation, mais uniquement par la rotation et se transforme en $\mathbf{b}' = \mathsf{R}\mathbf{b}$. La partie supérieure, le vecteur $\mathbf{a}$, représente la normale au plan formé par $\boldsymbol{L}$ et l'origine du repère. Lors de l'application d'un mouvement rigide, ce vecteur se transforme en $\mathbf{a}' = \mathsf{R}\mathbf{a} + [\mathbf{t}]_\wedge\mathsf{R}\mathbf{b}$. En rassemblant ces équations sous une forme matricielle, on obtient :

$$\mathbf{L}' \sim \begin{pmatrix} \mathsf{R} & [\mathbf{t}]_\wedge\mathsf{R} \\ 0 & \mathsf{R} \end{pmatrix} \mathbf{L}.$$

Nous renvoyons au travail de Navab et Faugeras [152] et au contenu de ce chapitre pour plus de détails sur la formulation de cette relation. On peut aisément imaginer une généralisation de cette matrice au cas d'une similitude, c'est à dire en incluant un facteur d'échelle. Mais peut-on généraliser au delà, c'est à dire avec une affinité ou une homographie ? Peut-on trouver une forme générale permettant de construire une telle matrice (6×6) pour tout mouvement usuellement représenté par une matrice (4×4) ?

Contributions. Nous proposons une représentation matricielle généralisée du mouvement permettant le transfert directe des droites en coordonnées de Plücker. Plus précisément, nous construisons les *matrices de mouvement pour droites 3D* (6×6), en fonction des matrices de mouvement usuelles (4×4). Nous formulons ces matrices puis en caractérisons les propriétés algébriques et géométriques via une approche stratifiée : nous étudions un cas de mouvement générique puis spécialisons nos résultats aux cas du mouvement en espaces projectif, affine, métrique et Euclidien. La spécialisation au cas Euclidien fait apparaître une matrice dont la structure est celle donnée par Navab et Faugeras [152]. Notre travail peut être regardé comme la généralisation de cette structure à tout type de mouvement.

Etant donnée une matrice (6×6), cette dernière doit respecter certaines *contraintes de consistance* pour représenter un mouvement. Nous formulons des jeux de contraintes minimaux pour les différents types de mouvement étudiés. Ces contraintes sont linéaires ou bilinéaires.

Nous étudions l'extraction de la matrice de mouvement usuelle à partir d'une matrice de mouvement pour droites 3D et développons des algorithmes pour les différents types de mouvement étudiés. Nous envisageons de même les cas où la matrice (6×6) considérée n'est pas une matrice de mouvement pour droites 3D, c'est à dire ne satisfait pas les contraintes de consistance. Finalement, nous illustrons la flexibilité de notre formulation en construisant une matrice de calibrage projectif vers métrique pour droites 3D.

La théorie développée dans ce chapitre est utilisée au chapitre 8, dans le cadre de l'alignement de reconstructions de droites.

Organisation du chapitre. Nous formulons les matrices de mouvement pour droites 3D pour différents types de mouvements, les contraintes de consistance et les propriétés algébriques associées en §7.2. Nous étudions une interprétation géométrique de ces matrices en §7.3 puis étudions l'extraction des paramètres de mouvement en §7.4. Finalement, nous formulons la matrice de passage projectif-métrique pour droites 3D en §7.5 et concluons en §7.6.

7.2 Les matrices de mouvement pour droites 3D

Nous formulons les matrices de mouvement pour droites 3D pour différents types de mouvement, en espace projectif, affine, métrique et Euclidien. Pour ce faire, nous considérons tout d'abord une matrice de mouvement usuelle générique et formons la *matrice de mouvement générique pour droites 3D*. A partir de cette formulation, la spécialisation aux différents types de mouvement mentionnés ci-dessus est effectuée par substitution.

7.2.1 Cas générique

Nous donnons la principale contribution de ce chapitre sous forme de la proposition suivante :

Proposition 1 *Les coordonnées de Plücker d'une droite 3D exprimées dans deux bases de l'espace différentes sont linéairement liées. La matrice* (6×6) $\tilde{\mathsf{G}}$ *que nous appelons* matrice de mouvement pour droites 3D *encapsule la transformation et peut être paramétrée par*[1] *:*

$$\tilde{\mathsf{G}} \sim \begin{pmatrix} \bar{\mathsf{G}}^{\star} & [\mathbf{g}_1]_{\wedge}\bar{\mathsf{G}} \\ -\bar{\mathsf{G}}[\mathbf{g}_2]_{\wedge} & g\bar{\mathsf{G}} - \mathbf{g}_1\mathbf{g}_2^{\mathsf{T}} \end{pmatrix}, \tag{7.1}$$

[1] La notation $\bar{\mathsf{G}}^{\star}$ représente la matrice des cofacteurs de la matrice $\bar{\mathsf{G}}$. Les propriétés de l'opérateur * sont étudiées en §2.3.

où G *est une matrice de mouvement usuelle* (4×4) *paramétrée par :*

$$\mathsf{G} \sim \begin{pmatrix} \bar{\mathsf{G}} & \mathbf{g}_1 \\ \mathbf{g}_2^\mathsf{T} & \bar{g} \end{pmatrix}. \tag{7.2}$$

Si $\mathbf{L}^\mathsf{T} \sim \begin{pmatrix} \mathbf{a}^\mathsf{T} & \mathbf{b}^\mathsf{T} \end{pmatrix}$ *sont des coordonnées de Plücker, c'est à dire si la contrainte de Plücker* $\mathbf{a}^\mathsf{T}\mathbf{b} = 0$ *est satisfaite, alors* $\tilde{\mathsf{G}}\mathbf{L}$ *sont les coordonnées de Plücker de la droite transférée, c'est à dire que* $\tilde{\mathsf{G}}\mathbf{L}$ *satisfait la contrainte de Plücker. Il n'y a pas de contrainte sur la matrice* G *: tout type de mouvement représenté par une matrice de mouvement usuelle* (4×4) *induit une matrice de mouvement pour droites 3D* (6×6).

7.2.1.1 Formulation (démonstration de la proposition 1)

Considérons une matrice de mouvement générique usuelle G paramétrée par l'équation (7.2). Pour formuler la matrice de mouvement générique pour droites 3D, considérons une correspondance de droites $\boldsymbol{L} \leftrightarrow \boldsymbol{L}'$. Soient deux points $\boldsymbol{M}$ et $\boldsymbol{N}$ sur $\boldsymbol{L}$, dont les coordonnées de Plücker sont :

$$\mathbf{L} \sim \begin{pmatrix} \mathbf{a} \\ \mathbf{b} \end{pmatrix} \sim \begin{pmatrix} \bar{\mathbf{M}} \wedge \bar{\mathbf{N}} \\ m\bar{\mathbf{N}} - n\bar{\mathbf{M}} \end{pmatrix}.$$

La droite $\boldsymbol{L}'$ est alors définie par deux points de coordonnées $\mathbf{M}' \sim \mathsf{G}\mathbf{M}$ et $\mathbf{N}' \sim \mathsf{G}\mathbf{N}$ et l'on peut écrire :

$$\begin{cases} \mathbf{M}' \sim \begin{pmatrix} \bar{\mathbf{M}}' \\ m' \end{pmatrix} \sim \begin{pmatrix} \bar{\mathsf{G}}\bar{\mathbf{M}} + m\mathbf{g}_1 \\ \mathbf{g}_2^\mathsf{T}\bar{\mathbf{M}} + gm \end{pmatrix} \\ \mathbf{N}' \sim \begin{pmatrix} \bar{\mathbf{N}}' \\ n' \end{pmatrix} \sim \begin{pmatrix} \bar{\mathsf{G}}\bar{\mathbf{N}} + n\mathbf{g}_1 \\ \mathbf{g}_2^\mathsf{T}\bar{\mathbf{N}} + gn \end{pmatrix}. \end{cases} \tag{7.3}$$

Formons les coordonnées de Plücker $\mathbf{L}'^\mathsf{T} \sim \begin{pmatrix} \mathbf{a}'^\mathsf{T} & \mathbf{b}'^\mathsf{T} \end{pmatrix}$ de la droite $\boldsymbol{L}'$. Considérons la partie supérieure, représentée par le vecteur-3 $\mathbf{a}'$. A partir de la définition (2.54), $\mathbf{a}' = \bar{\mathbf{M}}' \wedge \bar{\mathbf{N}}'$. Remplaçons $\bar{\mathbf{M}}'$ et $\bar{\mathbf{N}}'$ par leurs expressions données par l'équation (7.3) :

$$\begin{aligned} \mathbf{a}' &= \bar{\mathbf{M}}' \wedge \bar{\mathbf{N}}' \\ &= (\bar{\mathsf{G}}\bar{\mathbf{M}} + m\mathbf{g}_1) \wedge (\bar{\mathsf{G}}\bar{\mathbf{N}} + n\mathbf{g}_1) \\ &= (\bar{\mathsf{G}}\bar{\mathbf{M}}) \wedge (\bar{\mathsf{G}}\bar{\mathbf{N}}) + \underbrace{n(\bar{\mathsf{G}}\bar{\mathbf{M}}) \wedge \mathbf{g}_1}_{-n\mathbf{g}_1 \wedge (\bar{\mathsf{G}}\bar{\mathbf{M}})} + m\mathbf{g}_1 \wedge (\bar{\mathsf{G}}\bar{\mathbf{N}}) + mn\underbrace{\mathbf{g}_1 \wedge \mathbf{g}_1}_{\mathbf{0}}. \end{aligned}$$

D'après l'équation (2.2), $(\mathsf{S}\mathbf{x}) \wedge (\mathsf{S}\mathbf{y}) = \mathsf{S}^\star(\mathbf{x} \wedge \mathbf{y})$, où $\mathsf{S}^\star$ est la matrice des cofacteurs de S, ce qui conduit à l'expression suivante, linéaire en les coordonnées de Plücker $\mathbf{L} \sim \begin{pmatrix} \mathbf{a}^\mathsf{T} & \mathbf{b}^\mathsf{T} \end{pmatrix}$, identifiées ci-dessous d'après leur définition (2.54) :

$$\mathbf{a}' = \bar{\mathsf{G}}^\star(\underbrace{\bar{\mathbf{M}} \wedge \bar{\mathbf{N}}}_{\mathbf{a}}) + [\mathbf{g}_1]_\wedge \bar{\mathsf{G}}(\underbrace{m\bar{\mathbf{N}} - n\bar{\mathbf{M}}}_{\mathbf{b}}). \tag{7.4}$$

Considérons de même la partie inférieure $\mathbf{b}' = m'\bar{\mathbf{N}}' - n'\bar{\mathbf{M}}'$ des coordonnées de Plücker $\mathbf{L}'$. Remplaçons $\bar{\mathbf{M}}'$ et $\bar{\mathbf{N}}'$ par leurs expressions données par l'équation (7.3) :

$$\begin{aligned}\mathbf{b}' &= m'\bar{\mathbf{N}}' - n'\bar{\mathbf{M}}' \\ &= (\mathbf{g}_2^\mathsf{T}\bar{\mathbf{M}} + gm)(\bar{\mathsf{G}}\bar{\mathbf{N}} + n\mathbf{g}_1) - (\mathbf{g}_2^\mathsf{T}\bar{\mathbf{N}} + gn)(\bar{\mathsf{G}}\bar{\mathbf{M}} + m\mathbf{g}_1) \\ &= \mathbf{g}_2^\mathsf{T}\bar{\mathbf{M}}\bar{\mathsf{G}}\bar{\mathbf{N}} + n\mathbf{g}_2^\mathsf{T}\bar{\mathbf{M}}\mathbf{g}_1 + gm\bar{\mathsf{G}}\bar{\mathbf{N}} + gmn\mathbf{g}_1 - \mathbf{g}_2^\mathsf{T}\bar{\mathbf{N}}\bar{\mathsf{G}}\bar{\mathbf{M}} - m\mathbf{g}_2^\mathsf{T}\bar{\mathbf{N}}\mathbf{g}_1 - gn\bar{\mathsf{G}}\bar{\mathbf{M}} - gnm\mathbf{g}_1.\end{aligned}$$

Notre but est de factoriser cette expression pour faire apparaître les coordonnées de Plücker $\mathbf{L}$. Pour ce faire, nous regroupons les termes comme suit :

$$\mathbf{b}' = \underbrace{(\mathbf{g}_2^\mathsf{T}\bar{\mathbf{M}})\bar{\mathsf{G}}\bar{\mathbf{N}} - (\mathbf{g}_2^\mathsf{T}\bar{\mathbf{N}})\bar{\mathsf{G}}\bar{\mathbf{M}}}_{} + \underbrace{n\mathbf{g}_2^\mathsf{T}\bar{\mathbf{M}}\mathbf{g}_1 - m\mathbf{g}_2^\mathsf{T}\bar{\mathbf{N}}\mathbf{g}_1}_{} + \underbrace{gm\bar{\mathsf{G}}\bar{\mathbf{N}} - gn\bar{\mathsf{G}}\bar{\mathbf{M}}}_{}$$

Pour le premier terme, nous utilisons la propriété suivante : $(\mathbf{a}^\mathsf{T}\mathbf{b})\mathsf{C}\mathbf{d} - (\mathbf{a}^\mathsf{T}\mathbf{d})\mathsf{C}\mathbf{b} = -\mathsf{C}[\mathbf{a}]_\wedge(\mathbf{b}\wedge\mathbf{d})$ qui peut être démontrée en développant les deux côtés de l'équation. Nous obtenons alors :

$$(\mathbf{g}_2^\mathsf{T}\bar{\mathbf{M}})\bar{\mathsf{G}}\bar{\mathbf{N}} - (\mathbf{g}_2^\mathsf{T}\bar{\mathbf{N}})\bar{\mathsf{G}}\bar{\mathbf{M}} = -\bar{\mathsf{G}}[\mathbf{g}_2]_\wedge(\bar{\mathbf{M}}\wedge\bar{\mathbf{N}}) = -\bar{\mathsf{G}}[\mathbf{g}_2]_\wedge\mathbf{a}.$$

Nous réarrangeons le deuxième terme en $\mathbf{g}_2^\mathsf{T}(n\bar{\mathbf{M}} - m\bar{\mathbf{N}})\mathbf{g}_1$, ce qui donne :

$$\mathbf{g}_2^\mathsf{T}(n\bar{\mathbf{M}} - m\bar{\mathbf{N}})\mathbf{g}_1 = -(\mathbf{g}_2^\mathsf{T}\mathbf{b})\mathbf{g}_1 = -\mathbf{g}_1(\mathbf{g}_2^\mathsf{T}\mathbf{b}) = -\mathbf{g}_1\mathbf{g}_2^\mathsf{T}\mathbf{b}.$$

Le troisième terme se reécrit trivialement en :

$$gm\bar{\mathsf{G}}\bar{\mathbf{N}} - gn\bar{\mathsf{G}}\bar{\mathbf{M}} = g\bar{\mathsf{G}}(m\bar{\mathbf{N}} - n\bar{\mathbf{M}}) = g\bar{\mathsf{G}}\mathbf{b},$$

ce qui donne finalement la formule suivante pour $\mathbf{b}'$, linéaire en les coordonnées de Plücker $\mathbf{L}$:

$$\mathbf{b}' = -\bar{\mathsf{G}}[\mathbf{g}_2]_\wedge\mathbf{a} + (g\bar{\mathsf{G}} - \mathbf{g}_1\mathbf{g}_2^\mathsf{T})\mathbf{b}. \tag{7.5}$$

En regroupant les équations (7.4) et (7.5) sous une forme matricielle, nous obtenons la matrice de mouvement générique pour droites 3D, notée $\tilde{\mathsf{G}}$, telle qu'elle est définie par l'équation (7.1). ■

7.2.1.2 Propriétés algébriques

Nous explicitons certaines propriétés algébriques de la matrice de mouvement pour droites 3D. Ces dernières découlent de la formulation donnée par l'équation (7.1).

Définition à un facteur près. La matrice de mouvement usuelle est définie à un facteur multiplicatif non nul près. Il est donc logique que la matrice de mouvement pour droites 3D le soit aussi. Nous vérifions cela en construisant la matrice de mouvement pour droites 3D associée à la matrice de mouvement usuelle $\lambda\mathsf{G}$ où $\lambda \in \mathbb{R}^*$. Après un court calcul, nous obtenons la matrice $\lambda^2\tilde{\mathsf{G}}$, ce qui montre que le changement d'échelle d'une matrice de mouvement pour droites 3D ne change pas le mouvement représenté. De plus, la définition d'une matrice de mouvement pour droites 3D implique le transfert linéaire de coordonnées de Plücker : $\mathbf{L}' \sim \tilde{\mathsf{G}}\mathbf{L}$, ce qui entraîne $\mathbf{L}' \sim (\mu\tilde{\mathsf{G}})\mathbf{L}$, $\forall\mu \in \mathbb{R}^*$.

Analyse du nombre de degrés de liberté. La matrice de mouvement pour droites 3D définie par l'équation (7.1) dépend entièrement de la matrice de mouvement usuelle. Elle a donc le même nombre de paramètres et de degrés de liberté. Par exemple, la matrice d'homographie pour droites 3D a les 15 degrés de liberté induit par la matrice d'homographie usuelle.

Déterminant. Le déterminant de la matrice de mouvement pour droites 3D dépend directement de celui de la matrice de mouvement usuelle :

$$\det(\tilde{\mathsf{G}}) = (\det(\mathsf{G}))^3.$$

Cette propriété peut être vérifiée à l'aide d'un logiciel de calcul symbolique tel MAPLE.

Transposée. Soit B une matrice de mouvement et $\tilde{\mathsf{B}}$ la matrice de mouvement pour droites 3D associée. La propriété suivante est satisfaite :

$$(\mathsf{G} \sim \mathsf{B}^{\mathsf{T}}) \quad \Leftrightarrow \quad (\tilde{\mathsf{G}} \sim \tilde{\mathsf{B}}^{\mathsf{T}}).$$

Elle se démontre aisément en remplaçant dans l'équation (7.1) les valeurs de G et B. La notation $\tilde{\mathsf{B}}^{\mathsf{T}}$ peut donc être interprétée comme la transposée de la matrice $\tilde{\mathsf{B}}$ ou comme la matrice de mouvement pour droites 3D associée à la matrice B^{T}.

Inverse. Une propriété similaire au cas de la transposition est satisfaite dans le cas de l'inverse :

$$(\mathsf{G} \sim \mathsf{B}^{-1}) \quad \Leftrightarrow \quad (\tilde{\mathsf{G}} \sim \tilde{\mathsf{B}}^{-1}).$$

Cette propriété peut être vérifiée à l'aide d'un logiciel de calcul symbolique tel MAPLE. Similairement au cas de la transposition, on peut donc interpréter la notation $\tilde{\mathsf{B}}^{-1}$ comme l'inverse de la matrice $\tilde{\mathsf{B}}$ ou la matrice de mouvement pour droites 3D associée à la matrice B^{-1}.

7.2.2 Cas projectif

Nous avons considéré jusque-là une matrice de mouvement générique. Nous étudions maintenant la spécialisation des résultats obtenus à différents types de mouvement, avec une approche stratifiée (nous étudions successivement les cas des mouvements en espaces projectif, affine, métrique et Euclidien), et formulons des résultats spécifiques à chaque type de mouvement.

Dans cette section, nous considérons un type de mouvement homographique, représenté par une matrice de mouvement usuelle (4×4) notée H. Un tel type de mouvement se rencontre entre deux reconstructions projectives, c'est à dire obtenues à partir de caméras dont les paramètres internes sont inconnus.

7.2.2.1 Formulation

Nous formulons la matrice d'homographie pour droites 3D notée $\tilde{\mathsf{H}}$. Ce cas est le plus général car la matrice d'homographie H étant une matrice générale, on retrouve pour $\tilde{\mathsf{H}}$ la matrice de mouvement générique pour droites 3D donnée par l'équation (7.1), c'est à dire si l'on partitionne H comme dans l'équation (7.2) :

$$\tilde{\mathsf{H}} \sim \begin{pmatrix} \bar{\mathsf{H}}^{\star} & [\mathbf{h}_1]_{\wedge}\bar{\mathsf{H}} \\ -\bar{\mathsf{H}}[\mathbf{h}_2]_{\wedge} & h\bar{\mathsf{H}} - \mathbf{h}_1\mathbf{h}_2^{\mathsf{T}} \end{pmatrix}. \tag{7.6}$$

7.2.2.2 Contraintes de consistance

Nous désirons déterminer le nombre de contraintes de consistance devant être satisfaites par une matrice (6×6) $\tilde{\mathsf{B}}$ pour être une matrice d'homographie pour droites 3D valide, c'est à dire satisfaire la forme donnée par l'équation (7.6). Un mouvement homographique 3D a 15 degrés de liberté car il est représenté par une matrice (4×4) H définie à un facteur près. La matrice d'homographie pour droites 3D a donc aussi 15 degrés de liberté. Le nombre de contraintes recherché est donc de $6 \times 6 - 15 - 1 = 20$. Une contrainte est retirée car la matrice d'homographie pour droites 3D est définie à un facteur près. Ces contraintes sont non-linéaires en les coefficients de la matrice $\tilde{\mathsf{B}}$. Plus précisément, elles sont bilinéaires. Il existe plusieurs contraintes, seules 20 d'entre elles sont indépendantes. Nous donnons ci-dessous un jeu de 20 contraintes indépendantes. Pour ce

faire, nous utilisons le découpage suivant de la matrice $(6 \times 6)\tilde{\mathsf{B}}$ en sous-matrices (3×3) :

$$\tilde{\mathsf{B}} \sim \begin{pmatrix} \tilde{\mathsf{B}}_{11} & \tilde{\mathsf{B}}_{12} \\ \\ \tilde{\mathsf{B}}_{21} & \tilde{\mathsf{B}}_{22} \end{pmatrix}.$$

Notre formulation est basée sur l'hypothèse que l'homographie est représentée par une matrice telle que $\det(\bar{\mathsf{H}}) \neq 0$. Ceci implique que le rang de H est au moins 3. C'est le cas le plus important en pratique. Nous discutons ensuite les cas où $\det(\bar{\mathsf{H}}) = 0$.

Formulation. L'idée est de considérer le bloque $\tilde{\mathsf{B}}_{11}$ comme indépendant et de l'utiliser pour définir des contraintes sur les autres bloques. Notons qu'il est possible de formuler des contraintes similaires en partant du bloque $\tilde{\mathsf{B}}_{22}$. Nous formulons un premier groupe de 6 contraintes sur le bloque $\tilde{\mathsf{B}}_{12}$, en utilisant entre autres le fait que $[\mathbf{b}_1]_\wedge{}^\mathsf{T} = -[\mathbf{b}_1]_\wedge$, comme suit, :

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{12}^\mathsf{T} \;=\; -\bar{\mathsf{B}}^\star\bar{\mathsf{B}}^\mathsf{T}[\mathbf{b}_1]_\wedge \;=\; -\det(\bar{\mathsf{B}})[\mathbf{b}_1]_\wedge \;\sim\; [\mathbf{b}_1]_\wedge, \tag{7.7}$$

qui est une matrice anti-symétrique. Cela signifie que ses 3 coefficients diagonaux sont nuls et que ses 3 coefficients en position symétrique par rapport à la première diagonale ont des valeurs opposées.

Nous formulons de même un deuxième groupe de 6 contraintes sur le bloque $\tilde{\mathsf{B}}_{21}$ comme suit :

$$\tilde{\mathsf{B}}_{11}^\mathsf{T}\tilde{\mathsf{B}}_{21} \;=\; -\bar{\mathsf{B}}^{\mathsf{T}\star}\bar{\mathsf{B}}[\mathbf{b}_2]_\wedge \;=\; -\det(\bar{\mathsf{B}})[\mathbf{b}_2]_\wedge \;\sim\; [\mathbf{b}_2]_\wedge. \tag{7.8}$$

Finalement, nous formulons le troisième et dernier groupe de 8 contraintes sur le bloque $\tilde{\mathsf{B}}_{22}$ en considérant l'équation suivante :

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^\mathsf{T} + \tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{21}^\mathsf{T} \;=\; \bar{\mathsf{B}}^\star(b\bar{\mathsf{B}}^\mathsf{T} - \mathbf{b}_2\mathbf{b}_1^\mathsf{T}) + [\mathbf{b}_1]_\wedge\bar{\mathsf{B}}[\mathbf{b}_2]_\wedge\bar{\mathsf{B}}^\mathsf{T}.$$

Après développement et application de la formule (2.3) pour le dernier terme, nous obtenons :

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^\mathsf{T} + \tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{21}^\mathsf{T} \;=\; \det(\bar{\mathsf{B}})b\mathrm{I} - \bar{\mathsf{B}}^\star\mathbf{b}_2\mathbf{b}_1^\mathsf{T} + [\mathbf{b}_1]_\wedge[\bar{\mathsf{B}}^\star\mathbf{b}_2]_\wedge.$$

En utilisant la formule $-\mathbf{x}\mathbf{y}^\mathsf{T} + [\mathbf{y}]_\wedge[\mathbf{x}]_\wedge = -(\mathbf{x}^\mathsf{T}\mathbf{y})\mathrm{I}$ pour $\mathbf{x} = \bar{\mathsf{B}}^\star\mathbf{b}_2$ et $\mathbf{y} = \mathbf{b}_1$, nous obtenons :

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^\mathsf{T} + \tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{21}^\mathsf{T} \;=\; \det(\bar{\mathsf{B}})b\mathrm{I} - (\mathbf{b}_2^\mathsf{T}\bar{\mathsf{B}}^{\mathsf{T}\star}\mathbf{b}_1)\mathrm{I} \;=\; \left(\det(\bar{\mathsf{B}})b - (\mathbf{b}_2^\mathsf{T}\bar{\mathsf{B}}^{\mathsf{T}\star}\mathbf{b}_1)\right)\mathrm{I}. \tag{7.9}$$

Cette équation fournit 9 contraintes, une de trop. Cependant on observe que ces dernières résultent d'une égalité stricte, où l'échelle de la matrice $\tilde{\mathsf{B}}$ est prise en compte, ce que nous voulons éviter. Nous réecrivons donc l'équation (7.9) avec une égalité à un facteur près :

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^\mathsf{T} + \tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{21}^\mathsf{T} \;\sim\; \mathrm{I}. \tag{7.10}$$

A partir de cette équation, nous formulons les 8 contraintes recherchées : les 6 coefficients non diagonaux de $\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^\mathsf{T} + \tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{21}^\mathsf{T}$ sont nuls, ce qui donne 6 contraintes, et ses 3 coefficients diagonaux sont égaux, ce qui donne 2 contraintes.

Indépendance. Nous avons formulé un jeu de 20 contraintes, qui est donc minimal. D'autres contraintes pourraient avoir été formulées, par exemple en considérant l'anti-symétrie des produits $\tilde{\mathsf{B}}_{22}\tilde{\mathsf{B}}_{21}^\mathsf{T}$ et $\tilde{\mathsf{B}}_{22}^\mathsf{T}\tilde{\mathsf{B}}_{12}$. La question qui se pose alors naturellement est celle de l'indépendance des contraintes (7.7), (7.8) et (7.10). L'indépendance sera vérifiée si pour toute matrice $(6 \times 6)\,\tilde{\mathsf{B}}$ satisfaisant ces contraintes, il existe une unique matrice d'homographie usuelle $(4 \times 4)\,\mathsf{H}$ telle que $\tilde{\mathsf{H}} \sim \tilde{\mathsf{B}}$. Nous suivons un schéma constructiviste et exhibons une telle matrice H.

Considérons un partitionnement de H comme dans l'équation (7.2). Nous fixons tout d'abord $\bar{\mathsf{H}}^\star = \tilde{\mathsf{B}}_{11}$, et donc d'après l'équation (2.11) $\bar{\mathsf{H}} = \tilde{\mathsf{B}}^\star_{11}/\sqrt{|\det(\tilde{\mathsf{B}}_{11})|}$. Il est trivial de vérifier que $\tilde{\mathsf{H}}_{11} = \pm\tilde{\mathsf{B}}_{11}$ en recomposant $\tilde{\mathsf{H}}$. Les vecteurs $\mathbf{h}_1$ et $\mathbf{h}_2$ sont ensuite définis comme suit. D'après les contraintes (7.7) et (7.8), $\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}^\mathsf{T}_{12}$, ainsi que $\tilde{\mathsf{B}}^\mathsf{T}_{11}\tilde{\mathsf{B}}_{21}$ sont anti-symétriques. Nous choisissons $[\mathbf{h}_1]_\wedge = -\bar{\mathsf{H}}^{-1}\tilde{\mathsf{B}}^\mathsf{T}_{12}$ et $[\mathbf{h}_2]_\wedge = -\bar{\mathsf{H}}^{-\mathsf{T}}\tilde{\mathsf{B}}_{21}$. Il est trivial de voir que cette démarche garantit $\tilde{\mathsf{H}}_{12} = \pm\tilde{\mathsf{B}}_{12}$ et $\tilde{\mathsf{H}}_{21} = \pm\tilde{\mathsf{B}}_{21}$. Le choix est unique (au signe près). Concernant $\mathbf{h}_1$, c'est le noyau gauche du bloque $\tilde{\mathsf{B}}_{12} \sim [\mathbf{h}_1]_\wedge\bar{\mathsf{H}}$. Or $\bar{\mathsf{H}}$ est de rang plein par hypothèse. Pour $\mathbf{h}_1 \neq \mathbf{0}$, $[\mathbf{h}_1]_\wedge$ est de rang 2 ce qui implique que le noyau de $\tilde{\mathsf{B}}_{12}$ est de dimension 1. Pour $\mathbf{h}_1 = \mathbf{0}$, on trouve $[\mathbf{h}_1]_\wedge = 0$. Concernant $\mathbf{h}_2$, un raisonnement similaire montre l'unicité du choix.

Pour déterminer le scalaire h, nous nous basons sur le dernier groupe de contraintes. Telles qu'elles sont décrites par l'équation (7.10), ces contraintes ne permettent pas de déterminer h. La raison en est la suivante. Nous avons déjà déterminé 15 paramètres et nos contraintes sont choisies pour ne pas fixer l'échelle de $\tilde{\mathsf{B}}$. Les contraintes (7.7) et (7.8) aboutissant à une matrice anti-symétrique, elles sont intrinsèquement indépendantes de cette échelle, alors que les contraintes (7.10) n'en sont indépendantes que grâce à l'égalité à un facteur près. Nous utilisons donc l'équation (7.9) impliquant une égalité stricte. En remplaçant les valeurs extraites ci-dessus dans cette équation, nous obtenons :

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}^\mathsf{T}_{22} + \tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}^\mathsf{T}_{21} \;=\; \Big(\det(\bar{\mathsf{H}})h - (\mathbf{h}_2^\mathsf{T}\bar{\mathsf{H}}^{\mathsf{T}^\star}\mathbf{h}_1)\Big)\,\mathrm{I},$$

d'où h est extrait comme :

$$h = \frac{\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}^\mathsf{T}_{22} + \tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}^\mathsf{T}_{21} + (\mathbf{h}_2^\mathsf{T}\bar{\mathsf{H}}^{\mathsf{T}^\star}\mathbf{h}_1)\mathrm{I}}{\sqrt{3}\det(\bar{\mathsf{H}})}.$$

L'unicité est assurée car par hypothèse, le dénominateur n'est jamais nul. Nous avons donc construit d'une manière unique une matrice H telle que $\tilde{\mathsf{H}} = \pm\tilde{\mathsf{B}}$ et donc $\tilde{\mathsf{H}} \sim \tilde{\mathsf{B}}$, ce qui prouve l'indépendance des contraintes de consistance proposées. ■

Discussion sur le rang de la matrice B. Les contraintes (7.7), (7.8) et (7.10) ont été formulées sous l'hypothèse $\det(\bar{\mathsf{B}}) \neq 0$. Si on lève cette hypothèse, ces contraintes sont toujours valides, mais deviennent insuffisantes pour caractériser la forme de la matrice d'homographie pour droites 3D. Examinons par exemple le cas de la contrainte (7.7) :

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}^\mathsf{T}_{12} \;=\; -\bar{\mathsf{B}}^\star\bar{\mathsf{B}}^\mathsf{T}[\mathbf{b}_1]_\wedge \;=\; -\det(\bar{\mathsf{B}})[\mathbf{b}_1]_\wedge \;\sim\; [\mathbf{b}_1]_\wedge.$$

Lorsque $\det(\bar{\mathsf{B}}) = 0$, le produit $\tilde{\mathsf{B}}^\star\tilde{\mathsf{B}}^\mathsf{T}$ est nul. La contrainte est donc toujours valide car la matrice nulle est anti-symétrique. Par contre, cette contrainte est insuffisante pour l'extraction d'un unique vecteur $\mathbf{b}_1$. Les contraintes proposées sont donc nécessaires, mais non suffisantes lorsque $\det(\bar{\mathsf{B}}) = 0$.

Notons qu'une homographie définie par une matrice B telle que $\det(\bar{\mathsf{B}}) = 0$ transforme le plan à l'infini projectif (le plan d'équation $(0\ 0\ 0\ 1)^\mathsf{T}$) en un plan contenant l'origine. En effet, soit $\mathbf{q}$ le noyau de $\bar{\mathsf{B}}$ ($\bar{\mathsf{B}}\mathbf{q} = \mathbf{0}$), et soit $\boldsymbol{Q}$ le point 3D de coordonnées $\mathbf{Q}^\mathsf{T} = (\ \mathbf{q}^\mathsf{T}\quad 0)$. Le point $\boldsymbol{Q}$ se transforme en l'origine par B, car $\mathsf{B}\mathbf{Q} \sim (0\ 0\ 0\ 1)^\mathsf{T}$.

7.2.3 Cas affine

Nous considérons un mouvement de type affine, représenté par une matrice de mouvement usuelle (4×4) notée A et définie par :

$$\mathsf{A} \;\sim\; \begin{pmatrix} & \bar{\mathsf{A}} & & \mathbf{t} \\ & & & \\ 0 & 0 & 0 & a \end{pmatrix}, \tag{7.11}$$

avec $a \neq 0$. Un tel type de mouvement se rencontre entre deux reconstructions affines, c'est à dire où le plan à l'infini est identifié, mais où la conique absolue appartenant à ce plan est inconnue. Notons qu'il existe une formulation avec $a = 1$ encapsulant une contrainte de normalisation. L'utilisation de cette formulation sera discuté par la suite.

7.2.3.1 Formulation

Nous formulons la matrice d'affinité pour droites 3D notée $\tilde{\mathsf{A}}$ en remplaçant dans l'équation (7.1) la matrice de mouvement générique G par la matrice d'affinité A caractérisée par l'équation (7.11) et l'on obtient :

$$\tilde{\mathsf{A}} \sim \begin{pmatrix} \bar{\mathsf{A}}^\star & [\mathbf{t}]_\wedge \bar{\mathsf{A}} \\ \mathsf{0}_{(3\times 3)} & a\bar{\mathsf{A}} \end{pmatrix}. \tag{7.12}$$

7.2.3.2 Contraintes de consistance

Une affinité a 12 degrés de liberté. Une matrice (6×6) $\tilde{\mathsf{B}}$ doit donc satisfaire $6 \times 6 - 12 - 1 = 23$ contraintes pour être une matrice d'affinité pour droites 3D valide. Nous proposons un jeu de contraintes minimal. Comme dans le cas projectif, nous faisons l'hypothèse que la matrice d'affinité usuelle sous-jacente est telle que[2] $\det(\bar{\mathsf{A}}) \propto \det(\mathsf{A}) \neq 0$.

Un premier groupe de 9 contraintes est formulé trivialement sur $\tilde{\mathsf{B}}_{21}$:

$$\tilde{\mathsf{B}}_{21} = \mathsf{0}_{(3\times 3)}. \tag{7.13}$$

Un deuxième groupe de 8 contraintes se formule aussi directement sur $\tilde{\mathsf{B}}_{11}$ et $\tilde{\mathsf{B}}_{22}$:

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^\mathsf{T} = a\bar{\mathsf{A}}^\star\bar{\mathsf{A}}^\mathsf{T} = a(\det(\bar{\mathsf{A}}))\mathrm{I}. \tag{7.14}$$

Pour des raisons expliquées dans le cas projectif pour le groupe de contraintes données par l'équation (7.10), nous écrivons l'égalité à un facteur près pour qu'elle ne dépende pas de l'échelle de $\tilde{\mathsf{B}}$:

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^\mathsf{T} \sim \mathrm{I}. \tag{7.15}$$

Les 6 dernières contraintes se formulent en utilisant $\tilde{\mathsf{B}}_{12}$ et $\tilde{\mathsf{B}}_{11}$:

$$\tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{11}^\mathsf{T} = [\mathbf{t}]_\wedge \bar{\mathsf{A}}\bar{\mathsf{A}}^{\mathsf{T}\star} = \det(\bar{\mathsf{A}})[\mathbf{t}]_\wedge \sim [\mathbf{t}]_\wedge, \tag{7.16}$$

qui est anti-symétrique[3].

Indépendance des contraintes. Nous suivons un raisonnement constructiviste similaire au cas projectif : étant donnée une matrice $\tilde{\mathsf{B}}$ satisfaisant les contraintes (7.13), (7.15) et (7.16), le but est d'exhiber une matrice d'affinité usuelle A telle que $\tilde{\mathsf{A}} \sim \tilde{\mathsf{B}}$.

Nous fixons $\bar{\mathsf{A}}^\star = \tilde{\mathsf{B}}_{11}$, c'est à dire $\bar{\mathsf{A}} = \tilde{\mathsf{B}}_{11}^\star / \sqrt{|\det(\tilde{\mathsf{B}}_{11})|}$, d'après l'équation (2.11), ce qui implique $\tilde{\mathsf{A}}_{11} = \pm\tilde{\mathsf{B}}_{11}$. La translation $\mathbf{t}$ est déterminée comme suit. D'après la contrainte (7.16), $\tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{11}^\mathsf{T}$ est anti-symétrique. Nous fixons $[\mathbf{t}]_\wedge = \tilde{\mathsf{B}}_{12}\bar{\mathsf{A}}^{-1}$, ce qui implique $\tilde{\mathsf{A}}_{12} = \pm\tilde{\mathsf{B}}_{12}$. Finalement, nous utilisons l'équation (7.14) d'où l'on extrait $a = \|\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^\mathsf{T}\|/(\det(\bar{\mathsf{A}})\sqrt{3})$. Cette construction de la matrice A est unique pour des raisons similaires au cas projectif et garantit $\tilde{\mathsf{A}} = \pm\tilde{\mathsf{B}}$, ce qui démontre l'indépendance des contraintes proposées. ■

[2] Nous avons $\det(\mathsf{A}) = a\det(\bar{\mathsf{A}})$ et $a \neq 0$, d'où $\det(\bar{\mathsf{A}}) \propto \det(\mathsf{A})$.

[3] Des contraintes équivalentes, c'est à dire dépendantes, se formulent à partir de $\tilde{\mathsf{B}}_{12}$ et $\tilde{\mathsf{B}}_{22}$:

$$\tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{22}^{\mathsf{T}\star} = [\mathbf{t}]_\wedge \bar{\mathsf{A}}\bar{\mathsf{A}}^{\mathsf{T}\star} = \det(\bar{\mathsf{A}})[\mathbf{t}]_\wedge \sim [\mathbf{t}]_\wedge. \tag{7.17}$$

Il est trivial de voir que les contraintes (7.15) et (7.16) impliquant (7.17) et que (7.15) et (7.17) impliquant (7.16). Les contraintes (7.16) et (7.17) sont donc interchangeables.

Discussion sur le rang de la matrice B**.** Les contraintes (7.13), (7.15) et (7.16) ont été formulées sous l'hypothèse $\det(\mathsf{B}) \neq 0$. Comme dans le cas projectif, elles restent valide si l'on relâche cette hypothèse, mais ne sont plus suffisantes pour caractériser la structure de la matrice d'affinité pour droites 3D.

Discussion sur la normalisation $a = 1$**.** Une formulation courante de la matrice d'affinité usuelle définie par l'équation (7.11) est de fixer $a = 1$. Dans ce cas, la matrice A est définie à un facteur près, mais les bloques $\bar{\mathsf{A}}$ et $\mathbf{t}$ ne prennent individuellement un sens que lorsque la normalisation $A_{44} = 1$ est effectuée. L'extraction de ces bloques à partir de $\tilde{\mathsf{B}}$ nécessite donc de fixer l'échelle de cette dernière afin qu'une normalisation équivalente à $A_{44} = 1$ soit effective. Or, cette normalisation se traduit sous une forme plus complexe sur la matrice d'affinité pour droites 3D. Soit $\lambda \in \mathbb{R}^*$ le facteur de normalisation à appliquer à $\tilde{\mathsf{B}}$. On déduit de la paramétrisation (7.12) avec $a = 1$, $\lambda\tilde{\mathsf{B}}_{11} = (\lambda\tilde{\mathsf{B}}_{22})^\star$. En observant que d'après l'équation (2.4) $(\lambda\mathsf{S})^\star = \lambda^2\mathsf{S}^\star$, on obtient $\lambda\tilde{\mathsf{B}}_{11} = \lambda^2\tilde{\mathsf{B}}^\star_{22}$, d'où :

$$|\lambda| \quad = \quad \frac{\|\tilde{\mathsf{B}}_{11}\|}{\|\tilde{\mathsf{B}}^\star_{22}\|}.$$

7.2.4 Cas métrique

Nous considérons un mouvement de type similitude, représenté par une matrice de mouvement usuelle (4×4) notée S et définie par :

$$\mathsf{S} \quad \sim \quad \begin{pmatrix} & s\mathsf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{7.18}$$

Un tel type de mouvement se rencontre entre deux reconstructions métriques, c'est à dire où le calibrage des caméras est connu, mais où l'échelle globale n'a pas été fixée. La matrice R, le vecteur $\mathbf{t}$ et le scalaire s représentent respectivement la rotation, translation et le facteur d'échelle entre les deux reconstructions. Une similitude a 7 degrés de liberté. La matrice de rotation $\mathsf{R} \in SO(3)$ est soumise aux 6 contraintes d'orthonormalité suivantes $\mathsf{R}^\mathsf{T}\mathsf{R} = \mathrm{I}_{(3\times3)}$ et doit avoir un déterminant positif : $\det(\mathsf{R}) = 1$.

Dans cette section, nous parlerons de l'échelle de la transformation dans le sens du scalaire s et dans le sens du facteur d'échelle libre de la matrice S. Pour lever toute ambiguïté, nous emploierons les expressions "échelle géométrique" et "échelle algébrique" respectivement. La paramétrisation par l'équation (7.18) encapsule le facteur d'échelle algébrique par la normalisation $S_{44} = 1$. Dans le cas affine, nous avions choisi de paramétrer explicitement le coefficient $A_{44} = a$, voir équation (7.11). La raison, discutée dans la section précédente en est que la contrainte de normalisation $A_{44} = 1$ se traduit sous une forme non directe sur la matrice d'affinité pour droites 3D. Dans le cas métrique au contraire, la contrainte de normalisation $S_{44} = 1$ apparaît sous une forme simple dans la matrice de similarité pour droites 3D. Plus précisément, elle apparaît sous la forme de la normalisation du déterminant de la matrice de rotation.

7.2.4.1 Formulation

Nous formulons la matrice de similitude pour droites 3D notée $\tilde{\mathsf{S}}$ en remplaçant dans l'équation (7.1) la matrice de mouvement générique G par la matrice de similitude S caractérisée par l'équation (7.18) et l'on obtient :

$$\tilde{\mathsf{S}} \quad \sim \quad \begin{pmatrix} s\mathsf{R} & & [\mathbf{t}]_\wedge\mathsf{R} \\ & & \\ 0_{(3\times3)} & & \mathsf{R} \end{pmatrix}. \tag{7.19}$$

On voit resortir la matrice de rotation pour le bloque $\tilde{\mathsf{S}}_{22}$. C'est cette structure combinée à la contrainte $\det(\mathsf{R}) = 1$ qui permet de retrouver l'échelle algébrique et par suite l'échelle géométrique s.

7.2.4.2 Contraintes de consistance

La matrice de similitude pour droites 3D a les 7 degrés de liberté d'une similitude. Une matrice $(6 \times 6)\tilde{\mathsf{B}}$ doit donc satisfaire $6 \times 6 - 7 - 1 = 28$ contraintes pour être une matrice de similitude pour droites 3D valide.

Formulation. Le premier groupe de 9 contraintes est similaire au cas affine précédent :

$$\tilde{\mathsf{B}}_{21} \quad = \quad \mathsf{0}_{(3\times3)}. \tag{7.20}$$

L'orthonormalité de la matrice de rotation, bloque $\tilde{\mathsf{B}}_{22} = \mathsf{R}$ donne 6 contraintes, 5 sont des contraintes de consistance et une correspond à la normalisation $S_{44} = 1$ de l'équation (7.18). On obtient donc :

$$\tilde{\mathsf{B}}_{22}\tilde{\mathsf{B}}_{22}^{\mathsf{T}} \quad \sim \quad \mathrm{I}_{(3\times3)}. \tag{7.21}$$

Un troisième groupe de 8 contraintes se formule sur $\tilde{\mathsf{B}}_{11}$ et $\tilde{\mathsf{B}}_{22}$:

$$\tilde{\mathsf{B}}_{11}\tilde{\mathsf{B}}_{22}^{\mathsf{T}} \quad \sim \quad \mathrm{I}_{(3\times3)}. \tag{7.22}$$

Les 6 dernières contraintes sont obtenues entre $\tilde{\mathsf{B}}_{22}$ et $\tilde{\mathsf{B}}_{12}$:

$$\tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{22}^{\mathsf{T}} \quad \sim \quad [\mathbf{t}]_\wedge \mathsf{R}\mathsf{R}^{\mathsf{T}} \quad = \quad [\mathbf{t}]_\wedge \tag{7.23}$$

Un raisonnement similaire au cas affine montre que des contraintes dépendantes peuvent être dérivées du produit $\tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{11}^{\mathsf{T}}$.

Indépendance. Suivons l'approche constructiviste employée dans les cas précédents et exhibons une matrice de similarité usuelle S telle que $\tilde{\mathsf{S}} = \tilde{\mathsf{B}}$. Pour simplifier le raisonnement, nous commençons par fixer l'échelle algébrique de la matrice $\tilde{\mathsf{B}}$ en divisant par $\det(\tilde{\mathsf{B}}_{22})$. D'après la contrainte (7.21), $\tilde{\mathsf{B}}_{22}$ est alors une matrice de rotation et nous fixons $\mathsf{R} = \tilde{\mathsf{B}}_{22}$. D'après la contrainte (7.22), $\tilde{\mathsf{B}}_{11}$ est la même matrice de rotation à l'échelle près. Nous fixons $s = \frac{\|\tilde{\mathsf{B}}_{11}\|}{\|\tilde{\mathsf{B}}_{22}\|}$. D'après la contrainte (7.23), $\tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{22}^{\mathsf{T}}$ est anti-symétrique. Il est trivial de voir que le choix $[\mathbf{t}]_\wedge = \tilde{\mathsf{B}}_{12}\tilde{\mathsf{B}}_{22}^{\mathsf{T}}$ implique $\tilde{\mathsf{S}} = \tilde{\mathsf{B}}$. Des arguments similaires aux cas précédents assurent l'unicité. ■

7.2.5 Cas Euclidien

Nous considérons un mouvement rigide représenté par une matrice de mouvement usuelle (4×4) notée D et définie par :

$$\mathsf{D} \quad \sim \quad \begin{pmatrix} & \mathsf{R} & & \mathbf{t} \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{7.24}$$

Un tel type de mouvement se rencontre entre deux reconstructions Euclidiennes, c'est à dire obtenues avec des caméras calibrées, et dont l'échelle globale est identique.

La normalisation $D_{44} = 1$ ne pose pas de problème car elle est équivalente à la condition $\det(\mathsf{R}) = 1$ qui se retrouve également dans la matrice de mouvement rigide pour droites 3D.

7.2.5.1 Formulation

Nous formulons la matrice de mouvement rigide pour droites 3D notée $\tilde{\mathsf{D}}$ en remplaçant dans l'équation (7.1) la matrice de mouvement générique G par la matrice de mouvement rigide D caractérisée par l'équation (7.24). Notons qu'il est aussi possible de simplement remplacer l'échelle géométrique du cas de la similitude par la valeur $s = 1$ dans l'équation (7.19), ce qui donne la matrice (6×6), précédemment proposée par Navab et Faugeras [152] :

$$\tilde{\mathsf{D}} \sim \begin{pmatrix} \mathsf{R} & [\mathbf{t}]_\wedge \mathsf{R} \\ \mathsf{0}_{(3\times 3)} & \mathsf{R} \end{pmatrix}. \tag{7.25}$$

7.2.5.2 Contraintes de consistance

La matrice de mouvement rigide pour droites 3D a 6 degrés de liberté, 3 pour la rotation et 3 pour la translation. Une matrice (6×6) $\tilde{\mathsf{B}}$ doit donc satisfaire $6 \times 6 - 6 - 1 = 29$ contraintes pour être une matrice de mouvement rigide pour droites 3D valide. Ces contraintes sont très similaires à celles du cas précédent de la similitude. On retrouve les contraintes (7.20) (9 contraintes), (7.21) (5 contraintes) et (7.23) (6 contraintes). Les 9 dernières contraintes sont données par l'égalité des bloques $\tilde{\mathsf{B}}_{11}$ et $\tilde{\mathsf{B}}_{22}$. Prouver l'indépendance de ces contraintes en suivant un raisonnement constructiviste est trivial. ■

7.3 Une interprétation géométrique

Nous donnons une interprétation géométrique de combinaisons des bloques (3×3) composant les matrices de mouvement pour droites 3D. Pour ce faire, nous commençons par dresser un scénario de reconstruction avec deux jeux d'images de la même scène. Ce scénario est illustré sur la figure 7.1. Une reconstruction de droites et de caméras est obtenue à partir de chaque jeu d'images. Chaque reconstruction est exprimée dans une base de l'espace qui lui est propre. Nous parlerons de la première et de la deuxième base de reconstruction. Plus précisément, chaque reconstruction est exprimée telle qu'une caméra de référence ait la forme canonique suivante :

$$\mathsf{P} \sim \mathsf{P}' \sim (\mathrm{I} \quad \mathbf{0}),$$

où P et P' sont les matrices de projection des caméras de référence de chacun des jeux. Nous parlerons de la première et de la deuxième caméra de référence respectivement. La transformation entre ces deux reconstructions est représentée par une matrice de mouvement générique usuelle G telle que si $\mathbf{Q}$ sont les coordonnées d'un point dans la première base, alors $\mathsf{G}\mathbf{Q}$ sont ses coordonnées dans la deuxième base.

La notation P'' désigne la deuxième caméra de référence exprimée dans la première base. D'après la définition de G, nous avons :

$$\mathsf{P}'' \sim \mathsf{P}'\mathsf{G}.$$

7.3.1 Cas projectif

Nous développons une interprétation des bloques de la matrice d'homographie pour droites 3D $\tilde{\mathsf{H}}$, en remplaçant la matrice de mouvement générique G par une matrice d'homographie H dans le scénario décrit ci-dessus.

7.3.1.1 Scénario

Nous décrivons les particularités du scénario dans le cas projectif, illustré sur la figure 7.2. Une homographie ne conserve en général pas le plan à l'infini. Chaque reconstruction projective définit son propre plan à

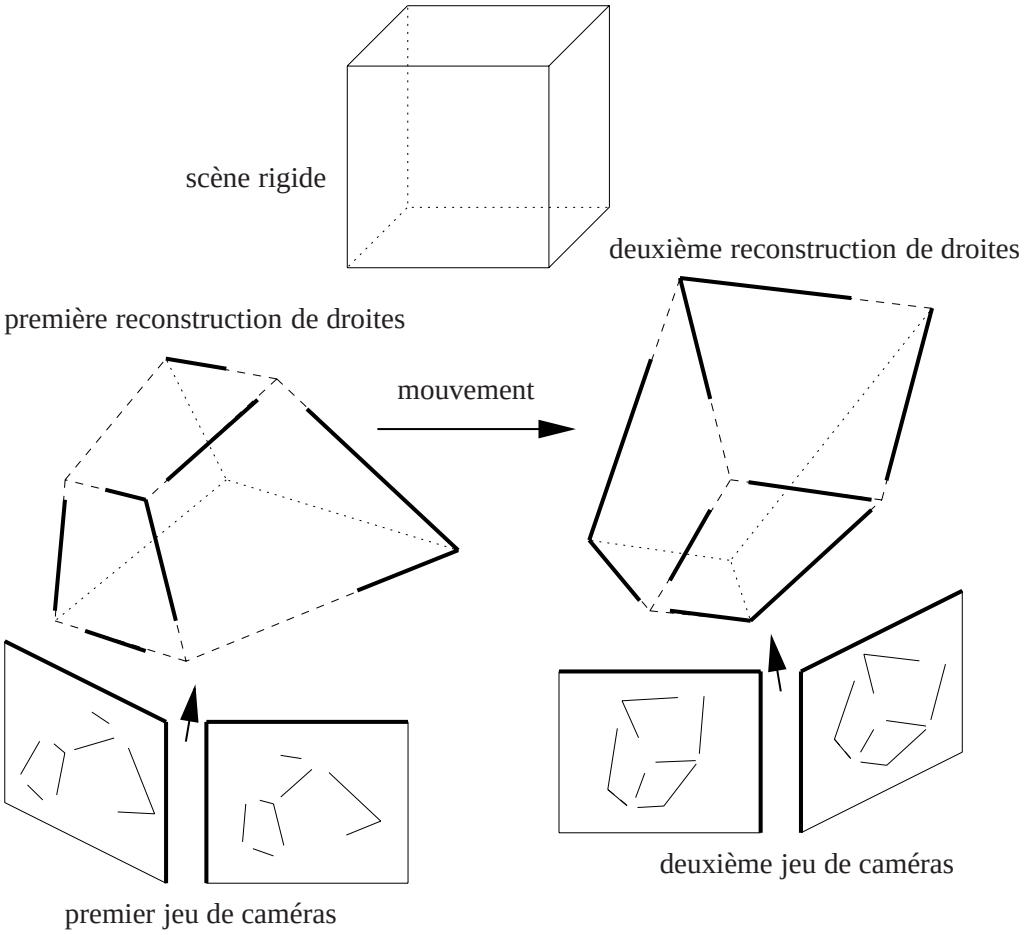


FIG. 7.1 – Le scénario considéré pour l'interprétation géométrique de combinaisons de bloques des matrices de mouvement pour droites 3D. Deux jeux d'images donnant lieu à deux reconstructions indépendantes de droites et de caméras sont considérées.

l'infini, de coordonnées $\mathbf{\Pi}_\infty \sim \mathbf{\Pi}'_\infty \sim (0\ 0\ 0\ 1)^\mathsf{T}$. Ces plans sont notés $\boldsymbol{\Pi}_\infty$ et $\boldsymbol{\Pi}'_\infty$ respectivement dans les deux reconstructions. Ils ne correspondent pas en général aux "vrais" plans à l'infini. Ils ne sont pas mis en correspondance par H. Nous notons $\mathbf{\Pi}''_\infty$ les coordonnées du plan $\boldsymbol{\Pi}'_\infty$ exprimées dans la première base :

$$\mathbf{\Pi}''_\infty \sim \mathsf{H}^\mathsf{T}\mathbf{\Pi}'_\infty.$$

Nous définissons les droites $\boldsymbol{L}_\infty \in \boldsymbol{\Pi}_\infty$ et $\boldsymbol{L}'_\infty \in \boldsymbol{\Pi}'_\infty$ comme deux droites reliées par H, ce qui équivaut à définir $\boldsymbol{L}_\infty = \boldsymbol{\Pi}_\infty \cap \boldsymbol{\Pi}'_\infty = \{\mathbf{Q} \sim (\ \bar{\mathbf{Q}}^\mathsf{T}\quad q)^\mathsf{T} \mid \bar{\mathbf{Q}}^\mathsf{T}\mathbf{h}_2 = q = 0\}$. En effet, tout point $\boldsymbol{Q} \in \boldsymbol{L}_\infty$ est sur $\boldsymbol{\Pi}_\infty$, donc $q = 0$, et se transforme sous l'action de H en un point à l'infini, d'où $\mathbf{h}_2^\mathsf{T}\bar{\mathbf{Q}} + hq = \mathbf{h}_2^\mathsf{T}\bar{\mathbf{Q}} = 0$.

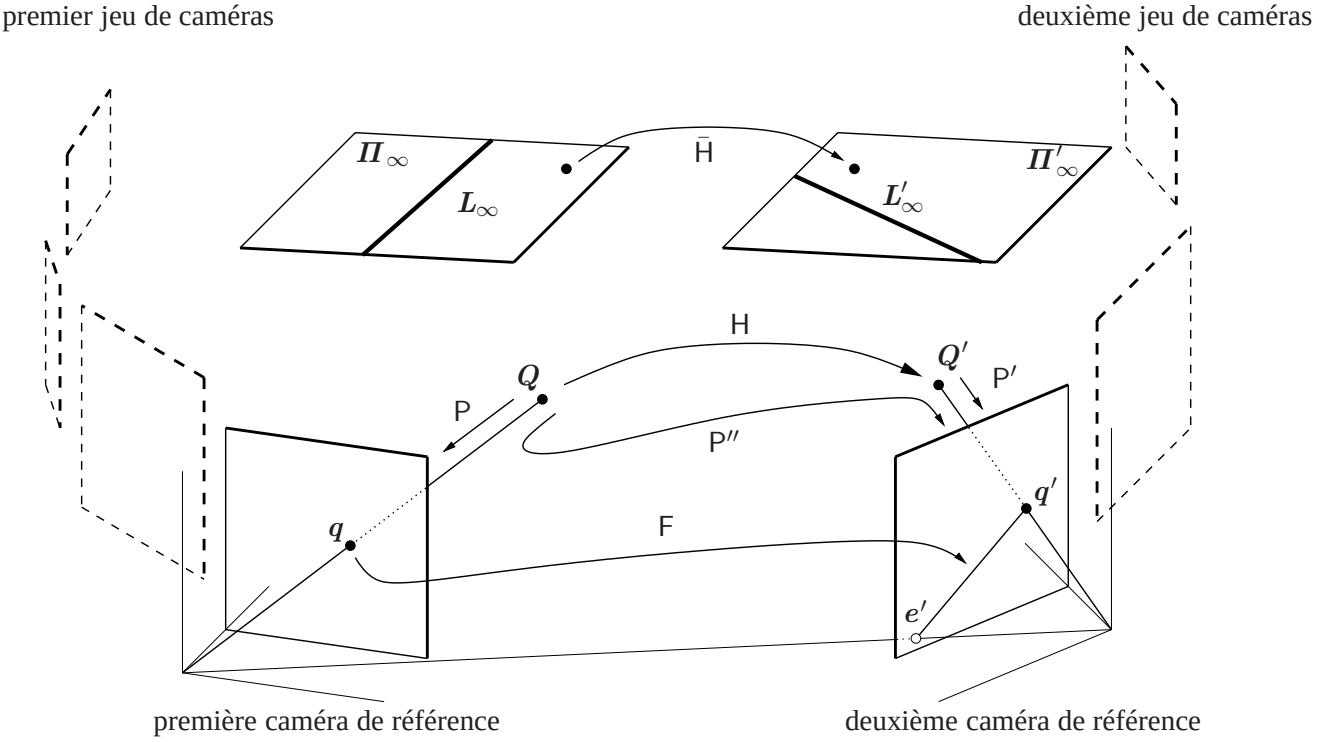


FIG. 7.2 – Le scénario considéré pour l'interprétation géométrique de combinaisons de bloques de la matrice d'homographie pour droites 3D. $\mathsf{P} \sim (\ \mathrm{I}\quad \mathbf{0})$ est la matrice de projection de la première caméra de référence et $\mathsf{P}' \sim (\ \mathrm{I}\quad \mathbf{0})$ de la deuxième caméra de référence, chacune exprimée dans la base de reconstruction qui lui est propre. $\mathsf{P}'' \sim \mathsf{P}'\mathsf{H}$ est la matrice de projection de la deuxième caméra de référence, exprimée dans la première base.

7.3.1.2 Interprétation de la matrice d'homographie usuelle

Nous commençons par donner une interprétation géométrique des composants de la matrice de mouvement usuelle H. Il est facile de vérifier que $(\ \bar{\mathsf{H}}\ \ \mathbf{h}_1) \sim \mathsf{P}''$. En effet :

$$\mathsf{P}'' \sim \mathsf{P}'\mathsf{H} \sim (\ \mathrm{I}\quad \mathbf{0}) \begin{pmatrix} \bar{\mathsf{H}} & \mathbf{h}_1 \\ \mathbf{h}_2^\mathsf{T} & h \end{pmatrix} \sim (\ \bar{\mathsf{H}}\quad \mathbf{h}_1). \tag{7.26}$$

Notons qu'il en découle alors que la matrice fondamentale entre les deux vues de référence est donnée par :

$$\mathsf{F} \sim [\mathbf{p}'']_\wedge \bar{\mathsf{P}}'' \sim [\mathbf{h}_1]_\wedge \bar{\mathsf{H}}. \tag{7.27}$$

Le bloque (3×3) $\bar{\mathsf{H}}$ est l'homographie de plan induite entre les vues de référence par le plan $\boldsymbol{\Pi}_\infty$. En effet, considérons un point $\boldsymbol{Q} \in \boldsymbol{\Pi}_\infty$. Les coordonnées de ce point s'écrivent $\mathbf{Q}^\mathsf{T} \sim \begin{pmatrix} \mathbf{q} & 0 \end{pmatrix}$, où $\mathbf{q}$ est le point correspondant dans la première vue de référence. Nous avons donc $\mathsf{P}'\mathsf{H}\mathbf{Q} \sim \bar{\mathsf{H}}\mathbf{q}$.

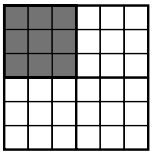
Le vecteur-3 $\mathbf{h}_1$ donne les coordonnées du second épipôle entre les deux vues de référence. En effet, d'après l'équation (7.27), $\mathsf{F}^\mathsf{T}\mathbf{h}_1 \sim \bar{\mathsf{H}}^\mathsf{T}[\mathbf{h}_1]_\wedge \mathbf{h}_1 = \mathbf{0}$.

Le vecteur-4 $\begin{pmatrix} \mathbf{h}_2^\mathsf{T} & h \end{pmatrix}^\mathsf{T}$ donne les coordonnées $\boldsymbol{\Pi}''_\infty$ du plan $\boldsymbol{\Pi}'_\infty$ de la deuxième reconstruction, exprimées dans la première base. En effet, nous avons $\boldsymbol{\Pi}''_\infty \sim \mathsf{G}^\mathsf{T}\boldsymbol{\Pi}'_\infty \sim \begin{pmatrix} \mathbf{g}_2^\mathsf{T} & g \end{pmatrix}^\mathsf{T}$.

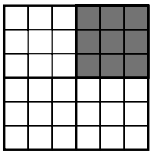
7.3.1.3 Interprétation de la matrice d'homographie pour droites 3D

Nous considérons différents bloques de la matrice d'homographie pour droites 3D $\tilde{\mathsf{H}}$, et donnons une interprétation géométrique de ces bloques, dans le cadre géométrique défini ci-dessus. Rappelons la définition et le partitionnement de $\tilde{\mathsf{H}}$:

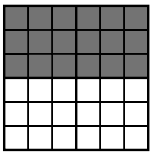
$$\tilde{\mathsf{H}} \sim \begin{pmatrix} \bar{\mathsf{H}}^\star & [\mathbf{h}_1]_\wedge \bar{\mathsf{H}} \\ -\bar{\mathsf{H}}[\mathbf{h}_2]_\wedge & h\bar{\mathsf{H}} - \mathbf{h}_1\mathbf{h}_2^\mathsf{T} \end{pmatrix} \sim \begin{pmatrix} \tilde{\mathsf{H}}_{11} & \tilde{\mathsf{H}}_{12} \\ \tilde{\mathsf{H}}_{21} & \tilde{\mathsf{H}}_{22} \end{pmatrix}.$$



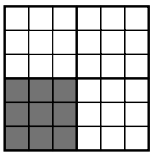
le bloque (3×3) $\tilde{\mathsf{H}}_{11}$ *est l'homographie de plan pour droites induite par le plan à l'infini entre les vues de référence.* En effet, lorsque $\det(\bar{\mathsf{H}}) \neq 0$, $\tilde{\mathsf{H}}_{11} \sim \bar{\mathsf{H}}^\star \sim \bar{\mathsf{H}}^{-\mathsf{T}}$, et nous avons observé que $\tilde{\mathsf{H}}_{11}^{-\mathsf{T}} \sim \bar{\mathsf{H}}$ était l'homographie de plan correspondante pour les points ;



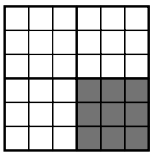
le bloque (3×3) $\tilde{\mathsf{H}}_{12}$ *est la matrice fondamentale entre les vues de référence*, c'est à dire $\tilde{\mathsf{H}}_{12} \sim [\mathbf{h}_1]_\wedge \tilde{\mathsf{H}} \sim \mathsf{F}$. Nous avons montré cette égalité, voir l'équation (7.27) ;



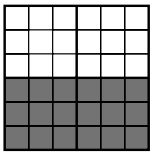
le bloque (3×6) $(\tilde{\mathsf{H}}_{11} \ \tilde{\mathsf{H}}_{12})$ *est la matrice de projection perspective pour droites en coordonnées de Plücker, pour la caméra de référence de la deuxième reconstruction exprimée dans la première base, c'est à dire* P'. En effet, $(\tilde{\mathsf{H}}_{11} \ \tilde{\mathsf{H}}_{12}) = (\bar{\mathsf{H}}^\star \ [\mathbf{h}_1]_\wedge \bar{\mathsf{H}}) \sim (\bar{\mathsf{P}}''^\star \ [\mathbf{p}'']_\wedge \bar{\mathsf{P}}'')$ d'après l'équation (7.26). Ce résultat correspond à la définition (6.3) de la matrice de projection perspective $\tilde{\mathsf{P}}''$ pour droites en coordonnées de Plücker ;



le bloque (3×3) $\tilde{\mathsf{H}}_{21}$ *est une homographie de plan dégénérée de point à droite entre les vues de référence.* Cette homographie peut être interprétée comme suit. Soit $\boldsymbol{l}$ une droite dans la première vue de référence. L'intersection de $\boldsymbol{l}$ et de la droite d'équation $\mathbf{h}_2$ est un point d'équation $\mathbf{q} \sim [\mathbf{h}_2]_\wedge \mathbf{l}$. La projection de $\boldsymbol{q}$ sur le plan à l'infini est sur $\boldsymbol{L}_\infty$. Elle est donnée par $\mathbf{Q}_\infty^\mathsf{T} \sim (\ \mathbf{q}^\mathsf{T} \quad 0)$. Le point correspondant dans la deuxième reconstruction est sur $\boldsymbol{L}'_\infty$, ses coordonnées sont données par $\mathbf{Q}'^\mathsf{T}_\infty \sim (\ \mathbf{q}'^\mathsf{T} \quad 0)$ avec $\mathbf{q}' \sim \bar{\mathsf{H}}[\mathbf{h}_2]_\wedge \mathbf{l}$. Projetons $\boldsymbol{q}'$ sur la deuxième vue de référence : on obtient $\mathbf{q}' \sim \bar{\mathsf{H}}[\mathbf{h}_2]_\wedge \mathbf{l} \sim \tilde{\mathsf{H}}_{21}\mathbf{l}$. Donc $\tilde{\mathsf{H}}_{21}$ est en un certain sens réciproque à la matrice fondamentale qui affecte des droites aux points. La matrice fondamentale donne des contraintes d'appariement pour les points, mais $\tilde{\mathsf{H}}_{21}$ ne donne aucune contrainte pour des droites générales ;



le bloque (3×3) $\tilde{\mathsf{H}}_{22}$ *est une homographie de plan pour points induite entre les vues de référence par* $\boldsymbol{\Pi}'_\infty$*, le plan à l'infini de la deuxième reconstruction.* Nous avons montré que $\bar{\mathsf{H}}$ était une homographie de plan et $\mathbf{h}_1$ le deuxième épipôle correspondant à la paire de vues de référence. En utilisant la formulation (2.51) de la famille des homographies de plan, $\tilde{\mathsf{H}}_{22} = h\bar{\mathsf{H}} - \mathbf{h}_1\mathbf{h}_2^\mathsf{T}$ est une homographie de plan induite par le plan $(\ \mathbf{h}_2^\mathsf{T} \quad h)^\mathsf{T} \sim \boldsymbol{\Pi}''_\infty$, qui sont les coordonnées du plan à l'infini de la deuxième reconstruction exprimées dans la première base ;



le bloque (3×6) $(\tilde{\mathsf{H}}_{21} \ \tilde{\mathsf{H}}_{22})$ *transfère une droite 3D avec les coordonnées de Plücker* $\mathbf{L}$ *de la première base vers la deuxième et projette son intersection avec* $\boldsymbol{\Pi}'_\infty$ *dans la deuxième vue de référence.* Nous pouvons voir ceci comme suit. Nous avons $\mathbf{Q}'^\mathsf{T}_\infty \sim (\ \mathbf{q}'^\mathsf{T} \quad 0)$ avec $\mathbf{q}' \sim (\tilde{\mathsf{H}}_{21} \ \tilde{\mathsf{H}}_{22})\mathbf{L}$, les coordonnées du point à l'infini de $\tilde{\mathsf{H}}\mathbf{L}$, d'après l'interprétation géométrique des coordonnées de Plücker, voir la section 2.12 du chapitre 2, et qui se projette en $\boldsymbol{q}'$ dans la deuxième vue de référence.

7.3.2 Cas affine

L'interprétation des bloques de la matrice d'affinité pour droites 3D est beaucoup plus facile et nous en donnons donc une description plus succinte. Le bloque $\tilde{\mathsf{A}}_{22} \sim \bar{\mathsf{A}}$ donne l'homographie à l'infini entre les vues de référence pour les points et $\tilde{\mathsf{A}}_{11} \sim \bar{\mathsf{A}}^\star$ pour les droites. Le bloque $\tilde{\mathsf{A}}_{12} \sim [\mathbf{t}]_\wedge \bar{\mathsf{A}}$ donne la matrice fondamentale entre les vues de référence comme dans le cas projectif.

7.3.3 Cas métrique et Euclidien

L'interprétation des bloques des matrices de similitude (7.19) et de mouvement rigide (7.25) pour droites 3D est triviale. Les bloques $\tilde{\mathsf{D}}_{11}$, $\tilde{\mathsf{D}}_{22}$ et $\tilde{\mathsf{S}}_{22}$ donnent la matrice de rotation (qui est l'homographie à l'infini pour des coordonnées calibrées pour le cas Euclidien). Le bloque $\tilde{\mathsf{S}}_{11}$ donne cette matrice de rotation multipliée par le carré du changement d'échelle. Les bloques $\tilde{\mathsf{D}}_{12}$ et $\tilde{\mathsf{S}}_{12}$ donnent la matrice essentielle entre les deux vues de référence.

7.4 Extraction du mouvement à partir d'une matrice de mouvement pour droites 3D

Etant donnée une matrice de mouvement pour droites 3D, il est possible d'en extraire les paramètres du mouvement correspondant, c'est à dire la matrice de mouvement usuelle (4×4) correspondante. Nous développons des algorithmes pour réaliser cette extraction dans les cas projectif, affine, métrique et Euclidien. Nous considérons le cas où la matrice (6×6) est exactement une matrice de mouvement pour droite 3D, c'est à dire que les contraintes de consistance développées en §7.2 sont satisfaites, et le cas où ces contraintes de consistance ne sont pas satisfaites. Dans ce cas, la structure de la matrice est bruitée et une solution approchée du mouvement est calculée. Une possibilité est de minimiser un critère aux moindres carrés basé sur la distance induite par la norme de Frobenius entre la matrice (6×6) donnée et la matrice de mouvement pour droites 3D correspondant aux paramètres estimés :

$$\min_{\mathsf{G}} \|\tilde{\mathsf{G}} - \tilde{\mathsf{B}}\|^2. \tag{7.28}$$

L'optimisation doit être effectuée sur une paramétrisation adaptée du mouvement considéré. Dû aux structures des matrices de mouvement pour droites 3D, c'est un problème d'optimisation non-linéaire que nous résolvons en utilisant un algorithme de type Levenberg-Marquardt. Dans les sections suivantes, nous développons des algorithmes pour le calcul d'une solution initiale. Ces algorithmes sont fortement liés aux preuves d'indépendance des contraintes de consistance de la section 7.2.

7.4.1 Cas projectif

Nous donnons dans le tableau 7.1 un algorithme permettant d'extraire une matrice d'homographie à partir d'une matrice d'homographie pour droites 3D, c'est à dire satisfaisant les contraintes de consistance (7.7), (7.8) et (7.10). Les notations $\pm$ et $\mp$ sont employées de manière consistante, c'est à dire $\pm x = - \mp x$.

1. $\bar{\mathsf{H}} = \pm\sqrt{|\det(\tilde{\mathsf{H}}_{11})|}\tilde{\mathsf{H}}_{11}^{-\mathsf{T}}$;
2. $[\mathbf{h}_1]_\wedge = \pm\tilde{\mathsf{H}}_{12}\bar{\mathsf{H}}^{-1}$ (détermination de $\mathbf{h}_1$) ;
3. $[\mathbf{h}_2]_\wedge = \mp\bar{\mathsf{H}}^{-1}\tilde{\mathsf{H}}_{21}$ (détermination de $\mathbf{h}_2$) ;
4. $h\mathrm{I}_{(3\times3)} = \pm(\tilde{\mathsf{H}}_{22} + \mathbf{h}_1\mathbf{h}_2^{\mathsf{T}})\bar{\mathsf{H}}^{-1}$ (détermination de h).

TAB. 7.1 – Algorithme d'extraction d'une matrice d'homographie H à partir d'une matrice d'homographie pour droites 3D $\tilde{\mathsf{H}}$.

Preuve : pour prouver la validité de cet algorithme, nous reformons une matrice d'homographie pour droites 3D $\tilde{\mathsf{H}}'$ à partir de la matrice d'homographie usuelle extraite, et vérifions que chacun de ses bloques correspond

au bloque original de $\tilde{\mathsf{H}}$:

$$\begin{aligned}
\tilde{\mathsf{H}}'_{11} &= \det(\bar{\mathsf{H}})\bar{\mathsf{H}}^{-\mathsf{T}} \\
&= \det\left(\pm\sqrt{|\det(\tilde{\mathsf{H}}_{11})|}\tilde{\mathsf{H}}_{11}^{-\mathsf{T}}\right)\frac{\pm 1}{\sqrt{|\det(\tilde{\mathsf{H}}_{11})|}}\tilde{\mathsf{H}}_{11} \\
&= \pm\sqrt{|\det(\tilde{\mathsf{H}}_{11})|}^{3}\frac{1}{\det(\tilde{\mathsf{H}}_{11})}\frac{\pm 1}{\sqrt{|\det(\tilde{\mathsf{H}}_{11})|}}\tilde{\mathsf{H}}_{11} \\
&= \pm\tilde{\mathsf{H}}_{11}
\end{aligned}$$

$$\begin{aligned}
\tilde{\mathsf{H}}'_{12} &= [\mathbf{h}_1]_\wedge\bar{\mathsf{H}} \\
&= \pm\tilde{\mathsf{H}}_{12}(\pm\bar{\mathsf{H}}^{-1})\bar{\mathsf{H}} \\
&= \pm\tilde{\mathsf{H}}_{12}
\end{aligned}$$

$$\begin{aligned}
\tilde{\mathsf{H}}'_{21} &= -\bar{\mathsf{H}}[\mathbf{h}_2]_\wedge \\
&= -\mp\bar{\mathsf{H}}(\pm\bar{\mathsf{H}}^{-1})\tilde{\mathsf{H}}_{21} \\
&= \pm\tilde{\mathsf{H}}_{21}
\end{aligned}$$

$$\begin{aligned}
\tilde{\mathsf{H}}'_{22} &= h\bar{\mathsf{H}} - \mathbf{h}_1\mathbf{h}_2^{\mathsf{T}} \\
&= \pm(\tilde{\mathsf{H}}_{22} + (\pm\mathbf{h}_1)(\pm\mathbf{h}_2^{\mathsf{T}}))(\bar{\mathsf{H}}^{-1})\bar{\mathsf{H}} - \mathbf{h}_1\mathbf{h}_2^{\mathsf{T}} \\
&= \pm\tilde{\mathsf{H}}_{22}. \blacksquare
\end{aligned}$$

Notons que l'algorithme du tableau (7.1) n'extrait l'homographie H qu'au signe près. Ce n'est pas un problème en pratique, l'échelle, et donc le signe de H ne sont pas significatifs. Ceci est lié au fait que, comme montré en §7.2.1.2, la matrice de mouvement pour droites 3D correspondant à $\lambda\mathsf{H}$ est $\lambda^2\tilde{\mathsf{H}}$: le signe du facteur λ est donc perdu.

En présence de bruit, c'est à dire lorsque les contraintes de consistance ne sont pas satisfaites, l'extraction est conduite au sens des moindres carrés comme indiqué précédemment. Nous proposons la solution d'initialisation suivante : au lieu de considérer la minimisation directe (7.28), nous minimisons une erreur aux moindres carrés à chaque étape de l'algorithme du tableau 7.1. D'autres solutions sont possibles. L'étape 1 est conservée telle quelle. Les étapes 2 et 3 requièrent l'ajustement d'une matrice anti-symétrique (3×3) $[\mathbf{w}]_\wedge$ à une matrice générale W. La solution suivante minimise la norme de Frobenius $\|[\mathbf{w}]_\wedge - \mathsf{W}\|$:

$$\mathbf{w}^{\mathsf{T}} = \frac{1}{2}(W_{32} - W_{23} \quad W_{13} - W_{31} \quad W_{21} - W_{12}). \tag{7.29}$$

L'étape 4 nécessite d'ajuster une matrice $\lambda\mathrm{I}$ proportionnelle à l'identité à une matrice générale (3×3) Λ. La solution suivante minimise la norme de Frobenius $\|\lambda\mathrm{I} - \Lambda\|$:

$$\lambda = \frac{1}{3}\sum_i \Lambda_{ii}. \tag{7.30}$$

7.4.2 Cas affine

Nous donnons un algorithme d'extraction dans le tableau 7.2 pour le cas où la matrice (6×6) considérée satisfait exactement les contraintes de consistance (7.13), (7.15) et (7.16). Pour prouver la validité de cet algorithme, nous renvoyons au cas projectif car ses 3 étapes en sont directement inspirées. Lorsque la matrice (6×6) donnée n'est pas exactement une matrice d'affinité pour droites 3D, nous proposons de modifier les étapes de l'algorithme du tableau 7.2 comme suit. L'étape 1 ne change pas. L'étape 2 peut être conduite comme dans le cas projectif, en utilisant l'équation (7.29), ainsi que l'étape 3, en utilisant l'équation (7.30).

1. $\bar{\mathsf{A}} = \pm\sqrt{|\det(\tilde{\mathsf{A}}_{11})|}\tilde{\mathsf{A}}_{11}^{-\mathsf{T}}$;
2. $[\mathbf{t}]_\wedge = \pm\tilde{\mathsf{A}}_{12}\bar{\mathsf{A}}^{-1}$ (détermination de $\mathbf{t}$) ;
3. $a\mathrm{I}_{(3\times3)} = \pm\tilde{\mathsf{A}}_{22}\mathsf{A}^{-1}$ (détermination de a).

TAB. 7.2 – Algorithme d'extraction d'une matrice d'affinité A à partir d'une matrice d'affinité pour droites 3D $\tilde{\mathsf{A}}$.

7.4.3 Cas métrique

Nous donnons un algorithme dans le tableau 7.3 pour le cas où la matrice (6×6) considérée satisfait exactement les contraintes de consistance. Prouvons la validité de cet algorithme. L'étape 1 de normalisation

1. normaliser $\tilde{\mathsf{S}}$ tel que $\det(\tilde{\mathsf{S}}_{22}) = 1$;
2. $s = \sqrt[3]{\det(\tilde{\mathsf{S}}_{11})}$;
3. $\mathsf{R} = \tilde{\mathsf{S}}_{22}$;
4. $[\mathbf{t}]_\wedge = \tilde{\mathsf{S}}_{12}\mathsf{R}^{\mathsf{T}}$ (détermination de $\mathbf{t}$).

TAB. 7.3 – Algorithme d'extraction d'une matrice de similitude S à partir d'une matrice de similitude pour droites 3D $\tilde{\mathsf{S}}$.

permet d'assurer que $\tilde{\mathsf{S}}_{22} = \mathsf{R}$ représente bien une matrice de rotation et donc que l'extraction de l'échelle s soit possible à partir de $\tilde{\mathsf{S}}_{11} = s\mathsf{R}$. L'étape 2 extrait l'échelle s :

$$\sqrt[3]{\det(\tilde{\mathsf{S}}_{11})} \;=\; \sqrt[3]{\det(s\mathsf{R})} \;=\; \sqrt[3]{s^3\det(\mathsf{R})} \;=\; s,$$

grâce à la normalisation effectuée par l'étape 1 ($\det(\mathsf{R}) = 1$). Les étapes 3 et 4 sont triviales.

Notons que, contrairement aux cas projectif et affine, il est possible de retrouver le signe de l'échelle. Ceci est dû au fait que la contrainte $\det(\mathsf{R}) = 1$ encapsule ce signe, contrairement à la contrainte de normalisation discutée dans le cas affine.

Pour le cas bruité, c'est à dire lorsque la matrice (6×6) ne satisfait pas exactement les contraintes de consistance, nous proposons la solution suivante. Les étapes 1 et 2 ne changent pas. L'étape 3 peut être effectuée en prenant en compte les deux versions de la rotation disponibles. On calcule alors $\hat{\mathsf{R}} = \frac{\tilde{\mathsf{D}}_{11}/s+\tilde{\mathsf{D}}_{22}}{2}$. Cette matrice doit être corrigée pour représenter une rotation, c'est à dire satisfaire les contraintes d'orthonormalité. Pour ce faire, nous calculons sa décomposition en valeurs singulières $\hat{\mathsf{R}} = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$ et affectons à R la valeur $\det(\mathsf{U})\det(\mathsf{V})\mathsf{U}\mathsf{V}^{\mathsf{T}}$ (le facteur $\det(\mathsf{U})\det(\mathsf{V})$ sert à assurer un déterminant positif à R). Ceci garantit que R est la matrice orthonormale la plus proche de $\hat{\mathsf{R}}$ au sens de la norme de Frobenius. L'étape 4 peut être conduite comme dans les cas projectif et affine, par la formule (7.29).

7.4.4 Cas Euclidien

Nous donnons un algorithme dans le tableau 7.4 pour le cas où la matrice (6×6) considérée satisfait exactement les contraintes de consistance. La preuve de cet algorithme est très proche du cas métrique (algorithme du tableau 7.3). De même, nous donnons une solution d'extraction très proche pour le cas bruité, consistant à utiliser la moyenne des deux versions de la matrice de rotation : $\hat{\mathsf{R}} = \frac{\tilde{\mathsf{D}}_{11}+\tilde{\mathsf{D}}_{22}}{2}$, puis à corriger le résultat pour qu'il satisfasse les contraintes d'orthonormalité : $\mathsf{R} = \mathsf{U}\mathsf{V}^{\mathsf{T}}$ où $\hat{\mathsf{R}} = \mathsf{U}\Sigma\mathsf{V}^{\mathsf{T}}$ est une décomposition en valeurs singulières. L'étape 3 est de même modifiée en se basant sur la formule (7.29).

1. normaliser $\tilde{\mathsf{D}}$ tel que $\det(\tilde{\mathsf{D}}_{11}) = 1$;
2. $\mathsf{R} = \tilde{\mathsf{D}}_{11}$;
3. $[\mathbf{t}]_\wedge = \tilde{\mathsf{D}}_{12}\mathsf{R}^\mathsf{T}$.

TAB. 7.4 – Algorithme d'extraction d'une matrice de mouvement rigide D à partir d'une matrice de mouvement rigide pour droites 3D $\tilde{\mathsf{D}}$.

7.5 La matrice de passage projectif-métrique pour droites 3D

La formulation des matrices de mouvement pour droites 3D par l'équation (7.1) est basée sur une matrice de mouvement usuelle générique. Cette formulation est très flexible car elle peut être spécialisée par simple substitution. Nous illustrons ceci en construisant une matrice de passage projectif-métrique pour droites 3D. Elle est basée sur une matrice de passage projectif-métrique usuelle H_u fournie par les algorithmes de calibrage en ligne, et permettant de convertir une reconstruction projective en reconstruction métrique. Plusieurs structures peuvent être utilisées pour la matrice de passage projectif-métrique usuelle. Si la reconstruction projective est exprimée dans le repère d'une caméra dont la matrice de calibrage est K et que le vecteur-4 $(\bar{\mathbf{\Pi}}^\mathsf{T} \quad \Pi)^\mathsf{T}$ représente les coordonnées du plan à l'infini dans la reconstruction projective, alors H_u prend la forme, voir par exemple [60, §7.4.2] :

$$\mathsf{H}_u \sim \begin{pmatrix} \mathsf{K}^{-1} & \mathbf{0}_{(3\times1)} \\ \bar{\mathbf{\Pi}}^\mathsf{T} & \Pi \end{pmatrix}. \tag{7.31}$$

En remplaçant dans l'équation (7.1) la matrice de mouvement générique pour droites 3D G par la matrice de passage projectif-métrique H_u définie par l'équation (7.31), on obtient la matrice de passage projectif-métrique pour droites 3D suivante :

$$\tilde{\mathsf{H}}_u \sim \begin{pmatrix} \frac{1}{f^2}\mathsf{K}^\mathsf{T} & \mathsf{0}_{(3\times3)} \\ -\mathsf{K}^{-1}[\bar{\mathbf{\Pi}}]_\wedge & \Pi\mathsf{K}^{-1} \end{pmatrix}. \tag{7.32}$$

7.6 Conclusions

Nous avons présenté les matrices de mouvement pour droites 3D (6×6), permettant d'appliquer un mouvement à une droite 3D exprimée en coordonnées de Plücker de manière linéaire. Ces matrices s'expriment directement en fonction des matrices de mouvement usuelles (4×4). Nous en donnons une formulation générique permettant de construire la matrice de mouvement pour droites 3D associée à une matrice de mouvement usuelle quelconque par simple substitution.

Nous employons une approche stratifiée et formulons successivement les matrices d'homographie, d'affinité, de similarité et de mouvement rigide pour droites 3D. Nous en étudions les propriétés algébriques et géométriques. Nous en caractérisons la structure en fournissant un jeu de contraintes minimales, satisfaites par toute matrice de mouvement pour droites 3D et telles que toute matrice (6×6) satisfaisant ces contraintes soit une matrice de mouvement pour droites 3D.

Nous développons des algorithmes d'extraction du mouvement à partir des matrices de mouvement pour droites 3D. Finalement, nous illustrons la flexibilité de notre formulation en construisant une matrice de cali-

brage pour droites 3D permettant de transférer une reconstruction projective de droites en une reconstruction métrique, dans un contexte de calibrage en ligne.

Une des applications de cette théorie est le calcul du mouvement 3D entre deux reconstructions 3D de droites, et l'alignement de ces reconstructions. Ce problème est examiné au chapitre suivant.

CHAPITRE

8

ALIGNEMENT DE RECONSTRUCTIONS DE DROITES

Considérons deux reconstructions indépendantes d'une scène rigide. Chacune est exprimée dans sa propre base de l'espace. Ces deux bases sont liées par une transformation géométrique. L'alignement des reconstructions consiste à les exprimer dans une base commune, en estimant cette transformation géométrique. Ce chapitre aborde le problème de l'alignement de reconstructions de droites à partir de correspondances de droites 3D.

Les difficultés principales sont dues au fait qu'il n'existe ni représentation algébrique naturelle, ni mesure de distance Euclidienne universelle pour droites 3D. La plupart des travaux existants considèrent des correspondances de points ou de segments, ou n'abordent que l'estimation d'une transformation rigide.

Nous adoptons une approche stratifiée, et examinons successivement l'alignement de reconstructions projectives, affines, métriques et Euclidiennes de droites. Le chapitre commence par une étude des cas minimaux, c'est à dire du nombre de droites nécessaire pour définir un mouvement unique. Nous formulons l'estimation au maximum de vraisemblance, basée sur la minimisation de l'erreur de reprojection. Les méthodes linéaires et quasi-linéaires proposées sont de deux types. Les méthodes basées sur une matrice de mouvement usuelle (4×4) traitent des données minimales et redondantes, mais minimisent une fonction de coût dont la forme est "duale" par rapport à celle de l'erreur de reprojection. Les méthodes basées sur une matrice de mouvement pour droites 3D (6×6) ne sont pas minimales. Elles minimisent une fonction de coût proche de l'erreur de reprojection, mais nécessitent une étape finale d'extraction du mouvement. Nous proposons ensuite des méthodes non-linéaires, minimisant différentes erreurs de transfert. Finalement, nous validons et comparons nos méthodes sur des données simulées et réelles.

Ces travaux ont été effectués en collaboration avec Peter Sturm, pour les méthodes basées sur une matrice de mouvement pour droites 3D et les méthodes non-linéaires, et ont été publiés dans [24, 20, 23], puis avec Richard Hartley et Fredrik Kahl, pour les méthodes basées sur une matrice de mouvement usuelle et le maximum de vraisemblance, et ont été publiés dans [19].

8.1 Introduction

Considérons deux reconstructions de la même scène rigide. Ces deux reconstructions, consistant d'un jeu de caméras et de primitives (points, droites, etc.), sont chacune exprimées dans leur propre repère de l'espace 3D. De tels repères sont, par exemple, centrés sur une caméra de référence de chaque reconstruction. L'alignement de reconstructions est équivalent au *calcul de la transformation 3D reliant ces deux repères*. Nous supposons que des correspondances de primitives sont connues. L'alignement de reconstructions est un problème important car il a de nombreuses applications, notamment la reconstruction à partir de longues séquences d'images et le calibrage en ligne de caméras. Plus précisément, c'est une des briques de base de toute une classe de méthodes de reconstruction multi-vues, voir en particulier les travaux de Fitzgibbon et Zisserman [66] et de Nistér [155]. Ces méthodes sont basées sur la fusion de reconstructions obtenues à partir de sous-séquences, ce qui permet une répartition de l'erreur sur toute la séquence. Chaque étape de fusion nécessite l'alignement de deux reconstructions.

La plupart des méthodes d'alignement sont basées sur des correspondances de points. De nombreux auteurs se concentrent sur l'estimation d'un mouvement rigide ou d'une similitude entre deux reconstructions obtenues avec des caméras calibrées, voir par exemple les travaux de Horn [101, 102] et de Arun et al. [5]. Csurka et al [47] et Zisserman et al. [237] étudient l'estimation d'une homographie entre deux reconstructions projectives de points. Ils proposent des méthodes linéaires et non-linéaires basées sur des contraintes 3D ou exprimées dans les images. L'utilisation de correspondances de plans est aussi possibles, car les équations linéaires qu'elles induisent sur le mouvement sont duales à celles données par les correspondances de points.

La droite est une primitive omniprésente en environnements humains, et dont la localisation dans les images est plus précise que celle des points. Nous renvoyons au chapitre 6 pour les avantages de l'utilisation de droites par rapport aux points. La manipulation de droites 3D est moins naturelle que celle des points et rend moins directe la tâche d'alignement. Ceci est dû à leur représentation algébrique, et au fait qu'il n'existe pas de mesure de distance Euclidienne entre droites, en 3D comme en 2D. Avant de pousser plus avant, précisons les termes droite et segment. Le terme *droite* désigne une entité infinie, par opposition au terme *segment*, qui désigne une entité bornée, composée par exemple d'une droite et de deux points sur cette droite.

La plupart des travaux existants abordent l'alignement de reconstructions métriques ou Euclidiennes. En outre, le cas des droites n'est que très rarement considéré. Par exemple, Zhang et Faugeras [235] proposent des méthodes à partir de correspondances de segments 3D. Faugeras et Herbert [59] considèrent des correspondances de droites 3D et proposent une méthode itérative dont la convergence n'est pas garantie. Kamgar-Parsi et al. [109] proposent des méthodes basées sur des correspondances 3D segment-segment, segment-droite et droite-droite. Ces méthodes ont un intérêt limité, car elles sont restreintes à l'estimation d'un mouvement rigide. Or, lorsqu'une reconstruction métrique est calculée, l'échelle globale reste indéterminée, tant que la mesure d'une distance réelle n'est pas connue. En pratique, on s'intéresse plutôt à l'estimation d'une similitude permettant de compenser le changement d'échelle, entre deux reconstructions métriques.

La localisation des segments et des droites dans les images est effectuée par ajustement à un ensemble de points. La direction d'un segment, c'est à dire en 2D, la droite sous-jacente, est déterminée de manière beaucoup plus fiable que ses extrémités, dont la localisation est très sensible aux conditions d'observation. Cet argument est une de nos motivations pour l'étude de l'alignement de reconstructions de droites. Ce chapitre a de forts liens avec les deux chapitres précédents. Les techniques développées au chapitre 6 sont utilisées pour la reconstruction de droites et de caméras à partir de correspondances, et l'étude algébrique et les algorithmes d'extraction proposés au chapitre 7, concernant les matrices de mouvement pour droites 3D, permettent la conception d'une partie de nos méthodes d'alignement.

Nous supposons que les droites 3D utilisées sont en position générale (il n'y pas de paire de droites coplanaire et de quintuplet formant un complexe linéaire ("Linear Line Complex" en Anglais), voir par exemple [189]), et en particulier, que les correspondances ne forment pas une configuration singulière pour l'estimation du mouvement 3D.

Contributions. Ce chapitre donne un ensemble très complet de méthodes d'alignement de reconstruction de droites, des méthodes minimales, pouvant être utilisées dans la boucle interne d'un estimateur robuste par

échantillonnage aléatoire tel RANSAC [64], en passant par des méthodes linéaires et non-linéaires minimisant diverses fonctions de coût, et l'estimateur au maximum de vraisemblance.

Nos solutions sont basées sur diverses fonctions de coût et modélisations algébriques du problème. Nous adoptons une approche stratifiée, et étudions successivement les cas de reconstructions projectives, affines, métriques et Euclidiennes. Nous commençons le chapitre par la détermination du nombre de correspondances de droites 3D nécessaire pour définir un mouvement unique dans ces différents espaces. C'est l'étude des *cas minimaux*. Nous déterminons aussi des cas minimaux basés sur des combinaisons de correspondances de droites et de points. Nous formulons le maximum de vraisemblance, dont la fonction de coût servira de modèle pour les estimateurs sous-optimaux présentés par la suite. Pour chaque espace cité ci-dessus, nous donnons des méthodes d'estimation "auto-initialisantes" (linéaires ou quasi-linéaires) et des méthodes non-linéaires, nécessitant une initialisation externe. Nous utilisons différentes modélisations du problème, basées sur une matrice de mouvement usuelle ou une matrice de mouvement pour droites 3D, et des fonctions de coût exprimées en 3D ou basées image. Nous comparons nos méthodes sur des données réelles ou simulées.

Organisation du chapitre. La section 8.2 étudie les cas minimaux. Nos notations, le scénario considéré et un résumé des différentes méthodes proposées sont donnés en §8.3. Nous formulons l'estimation au maximum de vraisemblance en §8.4. Nos méthodes d'estimation linéaires et quasi-linéaires sont établies en §8.5 avant l'examen des méthodes non-linéaires en §8.6. Nous donnons des résultats d'expériences en §8.7 puis nos conclusions et perspectives en §8.8.

8.2 Cas minimaux

Nous étudions les cas minimaux, c'est à dire le nombre de correspondances de droites minimal donnant assez de contraintes pour définir la transformation 3D recherchée. Chaque correspondance de droites donne 4 contraintes sur le mouvement. Le nombre minimal de correspondances pour chaque transformation 3D définie dans le tableau 2.1 – homographie, affinité, similitude et mouvement rigide – est donc donné en divisant le nombre de degrés de liberté de la transformation 3D par 4. Par exemple, en espace affine, il faut 3 correspondances de droites pour contraindre les 12 degrés de liberté de l'affinité. Cependant, en espace projectif, 4 correspondances ne suffisent pas à contraindre les 15 degrés de liberté de l'homographie. Une correspondance supplémentaire est nécessaire. Ce résultat découle de l'existence d'une famille d'isotropies de dimension 1, affectant tout groupe de 4 droites à lui-même, voir [46, 48, 81]. Cette famille d'isotropies n'existe pas en espaces affine, métrique et Euclidien. Le tableau 8.1 résume les cas minimaux. Le nombre de contraintes correspond strictement au nombre d'inconnues seulement dans le cas affine. Dans les autres cas, il est supérieur. Le tableau 8.1 donne des combinaisons de correspondances de droites et de points permettant d'obtenir exactement le nombre de contraintes nécessaires.

espace	mouvement	# ddl	# droites	minimale	# droites / # points
Euclidien	transformation rigide	6	2	non	-
métrique	similitude	7	2	non	1/1
affine	affinité	12	3	oui	-
projectif	homographie	15	5	non	3/1

TAB. 8.1 – Cas minimaux pour l'estimation d'un mouvement 3D à partir de correspondances de droites 3D. La colonne "# ddl" indique le nombre de degrés de liberté de la transformation, "# droites" indique le nombre de correspondances de droites minimum nécessaire pour l'estimation. La colonne "minimale" indique si les contraintes fournies par ces correspondances sont strictement minimales, c'est à dire si elles correspondent au nombre d'inconnues. Finalement, la colonne "# droites / # points" indique le nombre de correspondances de droites et de points minimal pour permettre l'estimation.

Finalement, nous faisons les deux remarques suivantes :

- ▷ en espace Euclidien, il n'existe pas de cas minimal, ni avec des correspondances de droites, ni avec des correspondances de points (2 correspondances de points ne permettent pas de déterminer la rotation autour de l'axe défini par chaque paire de points dans chaque base ; une troisième correspondance est donc requise) ;
- ▷ en espace métrique, le cas minimal avec 1 correspondance de point et de 1 correspondance de droite admet 4 solutions. En effet, toute configuration d'un point $\boldsymbol{Q}$ et d'une droite $\boldsymbol{L}$ admet pour isotropie une rotation d'angle π et dont l'axe est défini par le point $\boldsymbol{Q}$ et le point de $\boldsymbol{L}$ le plus proche de $\boldsymbol{Q}$.

8.3 Notations, scénario et désignation des méthodes proposées

Nous donnons nos notations conjointement à une description du scénario considéré. Nous étudions ensuite la formulation algébrique de l'estimation au maximum de vraisemblance, dont la fonction de coût sert de référence pour les méthodes sous-optimales proposées par la suite. Finalement, nous indiquons la notation utilisée pour désigner les différentes méthodes.

8.3.1 Notations et scénario

Nous considérons un scénario basé sur deux jeux de caméras indépendants. Les matrices de projection de ces caméras sont notées $\mathsf{P}_1, \ldots, \mathsf{P}_n$ pour le premier jeu et $\mathsf{P}'_1, \ldots, \mathsf{P}'_{n'}$ pour le deuxième. La notation avec un $'$ désigne le deuxième jeu d'images ou la deuxième reconstruction. Des droites sont localisées dans les images. La $j^{\text{ème}}$ droite dans l'image i est représentée par deux points notés $\boldsymbol{x}_{ij}$ et $\boldsymbol{y}_{ij}$. Notons que l'on peut, sans perte de généralité, représenter toute droite image définie comme le meilleur ajustement à un ensemble de points, par deux points et deux poids, voir le chapitre 3, §3.3.2. Les équations de ces droites sont notées $\boldsymbol{l}_{ij}$. Les droites sont reconstruites à partir des images, indépendamment pour chaque jeu de caméras. Chacune des deux reconstructions, notées $\boldsymbol{L}_1, \ldots, \boldsymbol{L}_m$ et $\boldsymbol{L}'_1, \ldots, \boldsymbol{L}'_m$, est exprimée dans un repère qui lui est propre. On suppose sans perte de généralité que le nombre de droites de chaque reconstruction, noté m, est identique, car seules les droites communes aux deux reconstructions sont utilisées pour l'alignement. Les coordonnées de Plücker des droites sont notées $\mathbf{L}_1, \ldots, \mathbf{L}_m$ et $\mathbf{L}'_1, \ldots, \mathbf{L}'_m$. Ces notations sont illustrées sur la figure 8.1.

8.3.2 Désignation et résumé des méthodes proposées

Le tableau 8.2 résume les différentes méthodes proposées. Les noms des méthodes sont construits comme suit. La nature de la transformation estimée est donnée par AFF pour une affinité ou SIM pour une similitude ou transformation rigide[1]. Si ces deux termes sont absents, alors la méthode est conçue pour estimer une homographie. LIN indique les méthodes linéaires, QLIN les méthodes quasi-linéaires et NLIN les méthodes non-linéaires, basées sur l'algorithme de Levenberg-Marquardt. Le préfixe 3DLMM indique une méthode basée sur une matrice de mouvement pour droites 3D. Si ce préfixe est absent, et que la méthode est linéaire ou quasi-linéaire, alors la méthode est basée sur une formulation avec une matrice de mouvement usuelle (4×4). Les méthodes non-linéaires sont valables pour tous les espaces, c'est à dire tous les types de mouvement. La mention 2D ou 3D indique une erreur basée image ou exprimée en 3D. Le maximum de vraisemblance est noté MLE. On utilise le caractère * pour dénoter un ensemble de méthodes, par exemple 3DLMM_* désigne toutes les méthodes basées sur une matrice de mouvement pour droites 3D.

8.4 Maximum de vraisemblance

8.4.1 Formulation algébrique

Nous donnons la fonction de coût correspondant au maximum de vraisemblance, désigné par MLE (de "Maximum Likelihood Estimator" en Anglais). La forme de cette fonction de coût sert de référence pour les

[1] Les méthodes sont proposées pour l'estimation d'une similitude et peuvent être spécialisées de manière triviale à l'estimation d'une transformation rigide.

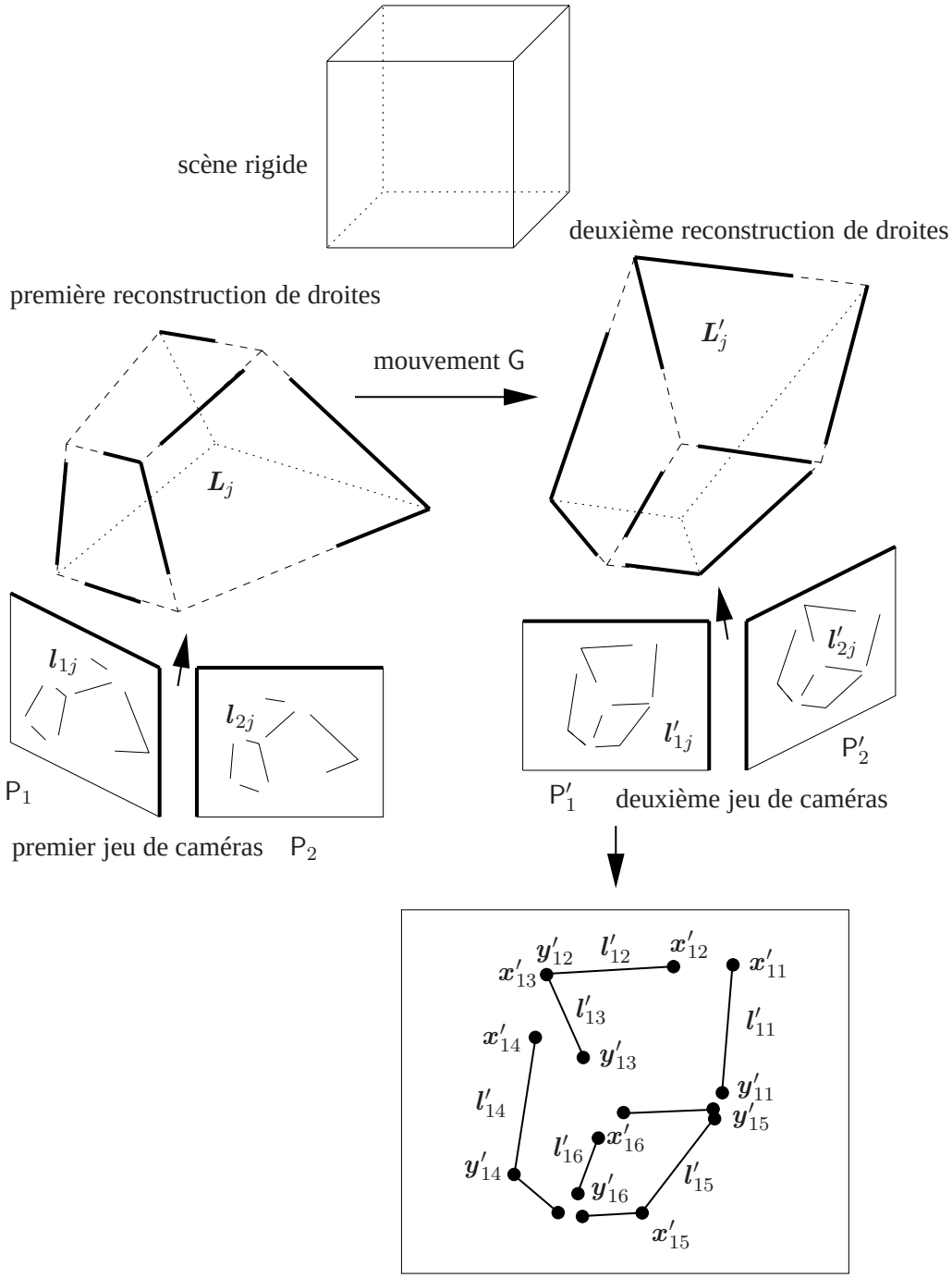


FIG. 8.1 – Le scénario considéré pour l'alignement de reconstructions de droites. Deux jeux de caméras indépendants, $\mathsf{P}_1, \ldots, \mathsf{P}_n$ et $\mathsf{P}'_1, \ldots, \mathsf{P}'_{n'}$ induisent deux reconstructions de la scène, notées $\boldsymbol{L}_1, \ldots, \boldsymbol{L}_m$ et $\boldsymbol{L}'_1, \ldots, \boldsymbol{L}'_m$. Ces deux reconstructions sont liées par une transformation 3D G inconnue.

méthode	résolution	type d'erreur résiduelle	sym.	# corresp.
estimation *linéaire* ou *quasi-linéaire* d'une *homographie*, §8.5.1				
LIN_3D	linéaire	3D, algébrique, point-plan	non	5
LIN_2D	linéaire	image, algébrique, point-droite	non	5
QLIN_2D	quasi-linéaire	image, Euclidienne, point-droite	non	5
3DLMM_LIN_3D	linéaire+extraction	3D, algébrique, droite-droite	non	7
3DLMM_LIN_2D_1	linéaire+extraction	image, algébrique, droite-droite	non	7
3DLMM_LIN_2D_2	linéaire+extraction	image, algébrique, droite-point	non	7
3DLMM_QLIN_2D	quasi-linéaire+extraction	image, Euclidienne, droite-point	non	7
estimation *linéaire* ou *quasi-linéaire* d'une *affinité*, §8.5.2				
AFF_LIN_3D	linéaire	3D, algébrique, point-plan	non	3
AFF_LIN_2D	linéaire	image, algébrique, point-droite	non	3
AFF_QLIN_2D	quasi-linéaire	image, Euclidienne, point-droite	non	3
3DLMM_AFF_LIN_3D	linéaire+extraction	3D, algébrique, droite-droite	non	4
estimation *linéaire* d'une *similitude* ou d'un *mouvement rigide*, §8.5.3				
3DLMM_SIM_LIN_3D	linéaire	3D, algébrique, droite-droite	non	2
estimation *non-linéaire*, pour *tout type de mouvement*, §§8.4 et 8.6				
NLIN_1	non-linéaire	image, Euclidienne, droite-point	non	-
NLIN_2	non-linéaire	image, Euclidienne, droite-point	oui	-
MLE	non-linéaire	image, Euclidienne, droite-point	oui	-

TAB. 8.2 – Résumé des méthodes proposées. La colonne "résolution" indique le type de résolution mis en œuvre ("extraction" signifie que la matrice de mouvement usuelle est extraite d'une matrice (6×6) en général proche d'une matrice de mouvement pour droites 3D), et la colonne "type d'erreur" résume les caractéristiques de la fonction de coût associée à la méthode (notons qu'une erreur "point-droite" est différente de "droite-point" : "point-droite" signifie une erreur mesurée entre des points prédits et des droites observées, et "droite-point" entre des droites prédites et des points mesurés). La colonne "sym." indique si la méthode confère un rôle symétrique aux deux jeux d'images, et la colonne "# corresp." donne le nombre de correspondances de droites minimum pour la méthode. Notons que seules les méthodes spécifiques à chaque espace sont mentionnées. Cela signifie que les méthodes proposées pour un certain espace peuvent être utilisées dans les espaces moins "généraux", par exemple les méthodes d'estimation d'une homographie peuvent être utilisées dans les autres espaces, en extrayant de l'homographie estimée le mouvement du type désiré.

autres méthodes : lorsque l'on conçoit une méthode sous-optimale linéaire ou non-linéaire, on porte un jugement a priori positif si la méthode induit une fonction de coût dont la forme est proche de celle du maximum de vraisemblance.

Pour définir algébriquement le maximum de vraisemblance, c'est à dire la fonction de coût, et sur quels paramètres doit être conduite l'optimisation, rappelons tout d'abord la formulation du maximum de vraisemblance pour la reconstruction de droites et de caméras. Ce problème, étudié en §6.4, chapitre 6, est formulé par la minimisation de l'erreur de reprojection, sur les paramètres de la structure (c'est à dire des droites reconstruites) $\mathcal{S} = \{\boldsymbol{L}_1, \ldots, \boldsymbol{L}_m\}$ et des caméras reconstruites $\mathcal{M} = \{\mathsf{P}_1, \ldots, \mathsf{P}_n\}$. Les droites image observées sont définies comme le meilleur ajustement à un ensemble de points. C'est sur la position de ces points que le bruit se manifeste, et c'est donc la somme des distances entre ces points et la droite reprojetée qui doit être minimisée. Cette somme de distances se réduit à la somme de distances entre la droite reprojetée et deux points particuliers, déterminés en fonction de tous les autres points, voir §3.3.2, chapitre 3. En notant $\hat{\boldsymbol{l}}_{ij}$ les droites reprojetées (c'est à dire prédites par le modèle) et $\boldsymbol{x}_{ij}$ et $\boldsymbol{y}_{ij}$ les points image, l'erreur de reprojection est donnée par :

$$\begin{aligned} \mathcal{L}(\mathcal{S}, \mathcal{M}) &= \sum_{j=1}^{m} \mathcal{L}(\boldsymbol{L}_j, \mathcal{M}) \\ &= \sum_{j=1}^{m} \sum_{i=1}^{n} \left(d^2_{PH}(\mathbf{x}_{ij}, \hat{\mathbf{l}}_{ij}) + d^2_{PH}(\mathbf{y}_{ij}, \hat{\mathbf{l}}_{ij}) \right). \end{aligned} \tag{8.1}$$

Dans le cas de l'estimation d'un mouvement, c'est l'erreur de reprojection dans les deux ensembles d'images qui doit être minimisée. On doit conduire l'optimisation sur le mouvement recherché et un ensemble de couples de droites 3D, exactement en correspondance par le mouvement. Si les positions des caméras ne sont pas parfaitement connues, elles peuvent être de même incluses dans la procédure d'optimisation. La fonction de coût est donnée par :

$$\mathcal{R}(\mathcal{S}, \mathcal{S}', \mathcal{M}, \mathcal{M}') = \mathcal{L}(\mathcal{S}, \mathcal{M}) + \mathcal{L}(\mathcal{S}', \mathcal{M}'), \tag{8.2}$$

où $\mathcal{S}$ et $\mathcal{S}'$ sont les paramètres de la structure des deux reconstructions et $\mathcal{M}$ et $\mathcal{M}$ les paramètres de chacun des deux jeux de caméras. Cette fonction de coût dépend du mouvement 3D recherché car les droites reconstruites doivent être parfaitement en correspondance. En pratique, on conduira la minimisation sur les paramètres de $\mathcal{S}$ et les paramètres du mouvement 3D. La structure $\mathcal{S}'$ dans la deuxième base est alors donnée en appliquant le mouvement à $\mathcal{S}$. Si l'on note $\mathcal{G}$ les paramètres du mouvement, et $\mathcal{G}(\mathcal{S})$ son application à la structure $\mathcal{S}$, alors, la structure au maximum de vraisemblance est donnée par :

$$\{\hat{\mathcal{S}}, \hat{\mathcal{G}}\} = \arg\min_{\mathcal{S},\mathcal{G}} \mathcal{R}(\mathcal{S}, \mathcal{G}(\mathcal{S}), \mathcal{M}, \mathcal{M}').$$

8.4.2 Optimisation non-linéaire

Le problème du maximum de vraisemblance est résolu en pratique par une optimisation non-linéaire basée sur l'algorithme de Levenberg-Marquardt, décrit en §3.6.2, chapitre 3, initialisée avec la solution fournie par un algorithme linéaire ou quasi-linéaire. La représentation orthonormale des droites 3D, décrite en §6.4.2, chapitre 6, est utilisée pour paramétrer la structure $\mathcal{S}$. Le mouvement 3D est paramétré par une matrice de mouvement usuelle (4×4) (la matrice de mouvement pour droites 3D correspondante est utilisée pour transférer les droites (voir ci-dessous) ; cependant, il est plus pratique de paramétrer la matrice de mouvement usuelle puis de former la matrice de mouvement pour droites 3D, que d'estimer directement les coefficients de cette dernière). Cette paramétrisation est spécialisée au type de mouvement estimé, par exemple dans le cas d'un mouvement rigide, on utilise un vecteur-3 pour la translation et une matrice de $SO(3)$ pour la rotation 3D, dont la mise à jour locale est effectuée avec 3 paramètres (les 3 angles d'Euler), comme décrit en §2.7.3, chapitre 2. La structure $\mathcal{S}'$ est déterminée par l'application du mouvement 3D à la structure $\mathcal{S}$. En pratique, on forme les coordonnées de Plücker des droites de la structure $\mathcal{S}$, auxquelles on applique la matrice de mouvement pour droites 3D induite

par la matrice de mouvement usuelle estimée, voir le chapitre 7, ce qui permet d'obtenir les coordonnées de Plücker des droites présentes dans la structure $\mathcal{S}$. Les caméras sont paramétrées comme décrit en §3.5.5, chapitre 3, c'est à dire que le repère de chaque reconstruction est centré sur la première caméra de chaque jeu. Les droites images prédites sont calculées en formant la matrice de projection (3×6) pour coordonnées de Plücker, voir §6.3.3, chapitre 6.

8.5 Méthodes linéaires et quasi-linéaires

Nous proposons différentes méthodes d'estimation "auto-initialisantes", c'est à dire ne nécessitant pas d'initialisation externe. Ces méthodes sont linéaires ou quasi-linéaires. Cette section est organisée d'une manière stratifiée : nous examinons tout d'abord le cas projectif, puis affine, en enfin les cas métrique et Euclidien. Deux catégories de méthodes apparaissent dans le cas projectif. Les méthodes basées sur l'estimation de la matrice d'homographie (4×4) usuelle et celles basées sur l'estimation de la matrice d'homographie (6×6) pour droites 3D. On retrouve ces deux catégories de méthodes dans le cas affine. Pour les cas métrique et Euclidien, on propose des méthodes spécifiques prenant en compte les contraintes non-linéaires d'orthonormalité sur les matrices de rotation, et des méthodes basées sur l'estimation d'une affinité puis l'extraction de la transformation recherchée.

Contrairement au maximum de vraisemblance, ces méthodes n'estiment que le mouvement. Elles minimisent une erreur de transfert, exprimée entre des primitives transférées de la première vers la deuxième reconstruction.

La différence fondamentale entre les méthodes basées sur une matrice de mouvement usuelle (4×4) et celles basées sur une matrice de mouvement pour droites 3D (6×6) réside dans la forme de la fonction de coût minimisée. Cette différence est similaire à celle observée au chapitre 6, pour la reconstruction de droites, entre l'algorithme de Hartley [92] et les algorithmes proposés : on montre en §6.3.1, chapitre 6, que l'algorithme de Hartley minimise une fonction de coût de forme "duale" à celle de l'erreur de reprojection. Ici, dans le cas d'une fonction de coût exprimée dans les images, les méthodes basées sur une matrice de mouvement usuelle minimisent une erreur "duale" à l'erreur de reprojection, exprimée à partir de distances entre des points prédits et des droites observées. Ces méthodes peuvent utiliser des données minimales ou redondantes et le résultat est obtenu directement. Elle peuvent facilement être combinées avec des correspondances de points ou de plans. Les méthodes basées sur une matrice de mouvement pour droites 3D minimisent une erreur de transfert exprimée avec des distances entre des droites prédites et des points observés, dont la forme est proche de celle de l'erreur de reprojection. La fonction de coût est donc de forme proche de celle du maximum de vraisemblance. Cependant, le mouvement, en espace projectif, n'est obtenu qu'après une phase d'extraction, introduisant un biais par rapport à la fonction de coût minimisée[2] En espaces affine, métrique et Euclidien, le calcul peut aussi être découpé en deux phases d'estimation. Ces méthodes ne sont pas minimales, et elles sont difficilement combinables avec d'autres types de correspondances.

8.5.1 Cas projectif

Comme introduit ci-dessus, nous proposons deux catégories de méthodes. La première est "directe" dans le sens où les coefficients de la matrice d'homographie usuelle sont estimés directement. Elle est basée sur une représentation différente des droites dans les deux bases. La deuxième catégorie de méthodes est basée sur l'estimation des coefficients d'une matrice d'homographie pour droites 3D, d'où est ensuite extraite la matrice d'homographie usuelle recherchée. Dû au bruit dans les données, la matrice (6×6) estimée ne satisfait pas exactement les contraintes de consistance formulées au chapitre 7, §7.2.2.2. L'étape d'extraction introduit donc un biais dans l'estimation, voir la méthode donnée en §7.4, chapitre 7.

[2] La phase d'extraction consiste à extraire une matrice d'homographie usuelle d'une matrice (6×6) estimée de manière linéaire, et en général proche d'une matrice d'homographie pour droites 3D.

8.5.1.1 Méthodes basées sur la matrice d'homographie usuelle

Nous proposons trois méthodes. La méthode LIN_3D est basée sur des contraintes exprimées en 3D et la méthode LIN_2D sur des contraintes basées image. Ces deux premières méthodes induisent des fonctions de coût basées sur des distances algébriques. La troisième méthode proposée, QLIN_2D, est la quasi-linéarisation de la méthode LIN_2D, et minimise une erreur exprimée à l'aide d'une distance Euclidienne.

L'idée clef de ces méthodes est de considérer une représentation différente pour chaque reconstruction de droites. Les droites de la première reconstruction sont représentées à l'aide de points, et celles de la deuxième par une représentation duale, des plans dans le cas des contraintes 3D, et une droite dans le cas des contraintes basées image. Cette idée exploite le fait que la matrice d'homographie usuelle permet un transfert linéaire des coordonnées homogènes des points.

Ces méthodes peuvent être facilement combinées avec des correspondances de points, dans le cas de données redondantes, ou pour des méthodes minimales, comme indiqué dans le tableau 8.1.

Contraintes 3D - méthode LIN_3D. Suivons l'idée évoquée ci-dessus et illustrée pour le cas 3D sur la figure 8.2. Considérons une correspondance de droites $\boldsymbol{L}_j \leftrightarrow \boldsymbol{L}'_j$, et notons H la matrice d'homographie usuelle

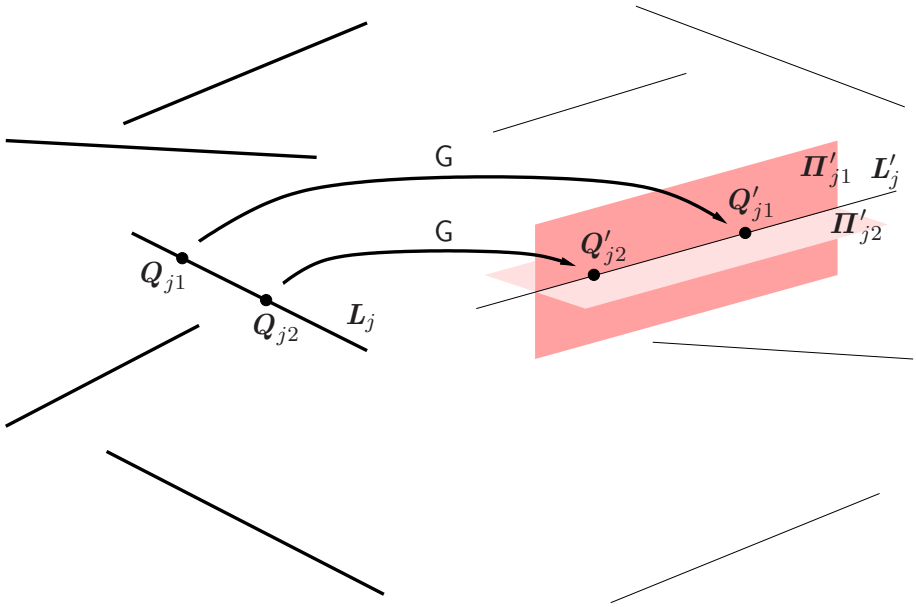


FIG. 8.2 – Le principe de la formulation de contraintes 3D linéaires sur les coefficients de la matrice de mouvement usuelle G : les droites de la première reconstruction sont représentées par des points et celle de la deuxième reconstruction par des plans. L'erreur est mesurée entre les points transférés de la première vers la deuxième base et les plans de la deuxième base.

inconnue reliant les deux reconstructions. Notons $\boldsymbol{Q}_{jk}$ un ensemble de points sur la droite $\boldsymbol{L}_j$. Les coordonnées homogènes des points correspondants (inconnus) $\boldsymbol{Q}'_{jk}$ dans la deuxième base peuvent être écrites comme une fonction linéaire de la matrice H : $\mathbf{Q}'_{jk} \sim \mathsf{H}\mathbf{Q}_{jk}$. Les droites $\boldsymbol{L}_j$ et $\boldsymbol{L}'_j$ étant correspondantes, on en déduit qu'en l'absence de bruit, $\boldsymbol{Q}'_{jk} \in \boldsymbol{L}'_j$. Nous désirons exprimer ces contraintes par des équations linéaires en les coefficients de la matrice H. Notons $\boldsymbol{\Pi}'_{jl}$ un ensemble de plans contenant $\boldsymbol{L}'_j$. Chacun des points $\boldsymbol{Q}'_{jk}$ est contenu sur chacun des plans $\boldsymbol{\Pi}'_{jl}$, ce qui s'exprime algébriquement par :

$$\boldsymbol{\Pi}'^{\mathsf{T}}_{jl}\mathsf{H}\mathbf{Q}_{jk} \;=\; 0. \tag{8.3}$$

Cette équation est linéaire en la matrice H. Un minimum de deux points ou plans est nécessaire pour représenter une droite 3D. Si nous écrivons la contrainte (8.3) pour $k = 1, 2$ et $l = 1, 2$, nous obtenons 4 équations, correspondant aux 4 contraintes fournies par une correspondance de droites sur le mouvement. En pratique,

nous utilisons plus de points et de plans afin de repartir l'erreur le long de la droite : plus les points sont bien répartis dans l'espace et plus les plans sont uniformément distribués autour de la droite, plus l'estimation est stable. Un choix possible pour les points est la reconstruction des extrémités des droites détectées dans les images, et pour les plans, les plans de vue de la droite, formés par la droite et les centres de projection des caméras.

Chaque équation peut être intégrée à un système homogène pour un vecteur-16 inconnu $\mathbf{h}$, défini comme la vectorisation par ligne de la matrice H. En supposant que $r \geq 2$ points et $s \geq 2$ plans sont utilisés, nous obtenons un ensemble de $rs \geq 4$ équations, dont seules 4 sont indépendantes. Pour m correspondances de droites, le système est donc de taille $(mrs \times 16)$[3] et s'écrit sous forme matricielle comme :

$$\mathsf{A}_{(mrs\times 16)}\mathbf{h}_{(16\times 1)} = \mathbf{0}_{(mrs\times 1)} \quad \text{avec} \quad \mathsf{A} \sim \begin{pmatrix} \cdots & & & \\ \Pi'_{jl,1}\mathbf{Q}_{jk}^{\mathsf{T}} & \Pi'_{jl,2}\mathbf{Q}_{jk}^{\mathsf{T}} & \Pi'_{jl,3}\mathbf{Q}_{jk}^{\mathsf{T}} & \Pi'_{jl,4}\mathbf{Q}_{jk}^{\mathsf{T}} \\ \cdots & & & \end{pmatrix}, \tag{8.4}$$

où $\mathbf{\Pi}_{jl}^{\mathsf{T}} = (\Pi_{jl,1} \;\; \Pi_{jl,2} \;\; \Pi_{jl,3} \;\; \Pi_{jl,4})$. En pratique, dû au bruit dans les mesures, le rang de la matrice A est 16. La solution pour $\mathbf{h}$ sous la contrainte $\|\mathbf{h}\|^2 = 1$ est alors donnée par le vecteur singulier associé à la plus petite valeur singulière de la matrice A, voir le chapitre 2, §2.6.1. La méthode proposée minimise la fonction de coût suivante :

$$\begin{aligned} \|\mathsf{A}\mathbf{h}\|^2 &= \cdots + (\mathbf{\Pi'}_{jl}^{\mathsf{T}}\mathsf{H}\mathbf{Q}_{jk})^2 + \cdots \\ &= \cdots + d^2_{A_PH}(\mathsf{H}\mathbf{Q}_{jk}, \mathbf{\Pi}'_{jl}) + \cdots , \end{aligned}$$

où d_{A_PH} est la distance algébrique point-plan définie par l'équation (2.19), dont la valeur dépend de l'échelle libre des coordonnées homogènes $\mathbf{Q}'_{jk}$ et $\mathbf{\Pi}'_{jl}$. La minimisation d'une erreur exprimée sur des quantités 3D n'a pas de sens physique en espace projectif.

Contraintes basées image - méthode LIN_2D. Nous suivons un raisonnement inspiré du cas 3D et illustré sur la figure 8.3. Nous utilisons un ensemble de points $\boldsymbol{Q}_{jk} \in \boldsymbol{L}_j$. Le but est d'exprimer les contraintes $\boldsymbol{Q}'_{jk} \in \boldsymbol{L}'_j$ avec $\mathbf{Q}'_{jk} \sim \mathsf{H}\mathbf{Q}_{jk}$, par des équations linéaires en la matrice H. La différence avec le cas 3D précédent est que nous désirons exprimer ces contraintes basées sur des entités mesurées dans les images. Considérons le deuxième jeu de caméras. La projection perspective préserve les relations d'incidence : les projections des points $\boldsymbol{Q}'_{jk}$ se trouvent donc sur les images de $\boldsymbol{L}'_j$. Notons $\mathsf{P}'_{i'}$ les matrices de projection du deuxième jeu de caméras et $\boldsymbol{l}'_{i'j}$ les images correspondantes de la droite $\boldsymbol{L}'_j$. Les points transférés puis reprojetés s'écrivent $\hat{\mathbf{q}}'_{i'jk} \sim \mathsf{P}'_{i'}\mathsf{H}\mathbf{Q}_{jk}$. La contrainte $\hat{\boldsymbol{q}}'_{i'jk} \in \boldsymbol{l}'_{i'j}$ s'exprime algébriquement par :

$$\mathrm{l}'^{\mathsf{T}}_{i'j}\mathsf{P}'_{i'}\mathsf{H}\mathbf{Q}_{jk} = 0. \tag{8.5}$$

Cette équation est linéaire en la matrice H. Le nombre minimum d'images pour reconstruire une droite 3D est deux. Si nous écrivons la contrainte (8.3) pour $k = 1, 2$ et $\acute{\imath} = 1, 2$, nous obtenons 4 équations, correspondant aux 4 contraintes fournies par une correspondance de droites sur le mouvement. En pratique, comme dans le cas 3D, nous utilisons plus de points et d'images afin de repartir l'erreur le long de la droite. Pour les droites image, un choix naturel est d'utiliser toutes les images disponibles, permettant de repartir l'erreur minimisée sur toutes ces dernières.

Chaque équation peut être intégrée à un système homogène pour le vecteur-16 inconnu $\mathbf{h}$, défini comme dans le cas 3D par la vectorisation par ligne de la matrice H. En supposons que $r \geq 2$ points et $\acute{n} \geq 2$ droites images sont utilisés, nous obtenons un ensemble de $r\acute{n} \geq 4$ équations, dont seules 4 sont indépendantes. Pour m correspondances de droites, le système est donc de taille $(mr\acute{n} \times 16)$[4] et s'écrit sous forme matricielle

[3] Le nombre d'équations mrs suppose que le même nombre de points et de plans est utilisé pour chaque correspondance de droites. Cette hypothèse n'est utilisée qu'à des fins d'allègement des notations et n'est pas du tout essentielle au raisonnement proposé.

[4] Comme dans le cas 3D, le nombre d'équations mrn' suppose que le même nombre de points et de droites images est utilisé pour représenter chaque correspondance de droites. Cette hypothèse n'est utilisée qu'à des fins d'allégement des notations et n'est pas du tout essentielle au raisonnement proposé.

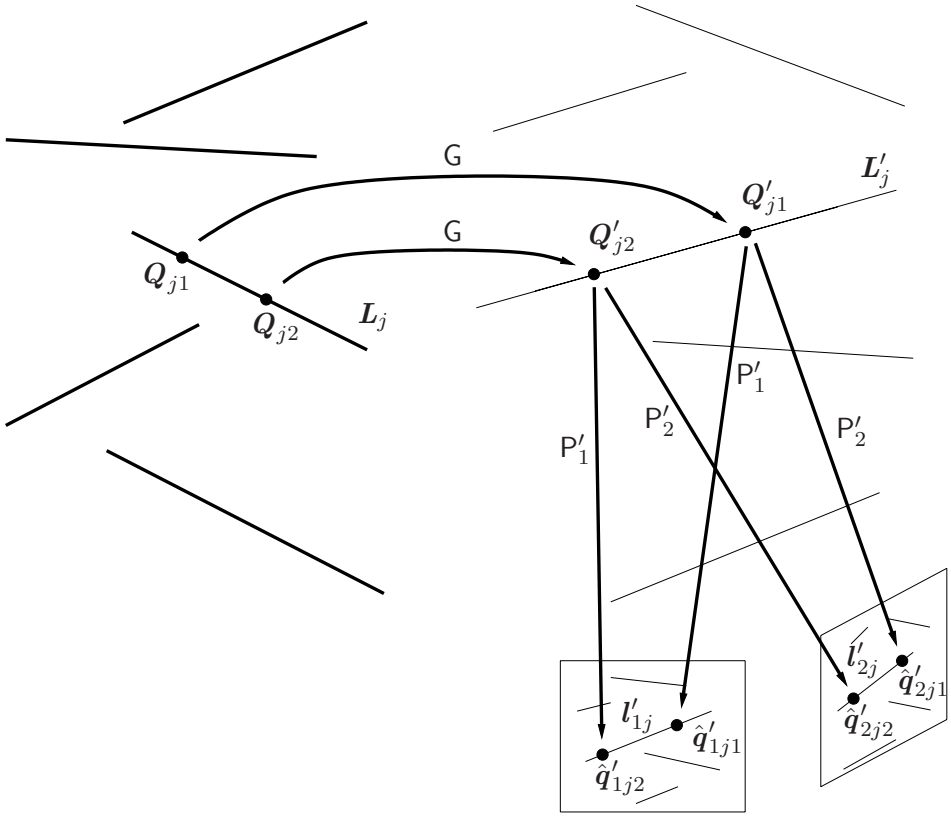


FIG. 8.3 – Le principe de la formulation de contraintes basées image linéaires sur les coefficients d'une matrice de mouvement usuelle G : les droites de la première reconstruction sont représentées par des points et celle de la deuxième reconstruction par des droites images. L'erreur est mesurée entre les points transférés de la première vers la deuxième base et reprojetés dans le deuxième jeu d'images et les droites observées dans le deuxième jeu d'images.

comme :

$$\mathsf{A}_{(mrn' \times 16)}\mathbf{h}_{(16\times 1)} \quad = \quad \mathbf{0}_{(mrn' \times 1)},$$

avec :

$$\mathsf{A} \quad \sim \quad \begin{pmatrix} & \cdots & & \\ (\mathbf{l}'^{\mathsf{T}}_{i'j}\mathbf{p}'_{i',1})\mathbf{Q}^{\mathsf{T}}_{jk} & (\mathbf{l}'^{\mathsf{T}}_{i'j}\mathbf{p}'_{i',2})\mathbf{Q}^{\mathsf{T}}_{jk} & (\mathbf{l}'^{\mathsf{T}}_{i'j}\mathbf{p}'_{i',3})\mathbf{Q}^{\mathsf{T}}_{jk} & (\mathbf{l}'^{\mathsf{T}}_{i'j}\mathbf{p}'_{i',4})\mathbf{Q}^{\mathsf{T}}_{jk} \\ & \cdots & & \end{pmatrix}, \tag{8.6}$$

où les vecteurs-3 $\mathbf{p}'_{i',c}$ désignent les colonnes de la matrice $\mathsf{P}'_{i'}$. En pratique, dû au bruit dans les mesures, le rang de la matrice A est 16. Nous minimisons l'erreur $\|\mathsf{A}\mathbf{h}\|^2$ sous la contrainte $\|\mathbf{h}\|^2 = 1$. La solution pour $\mathbf{h}$ est alors donnée par le vecteur singulier associé à la plus petite valeur singulière de la matrice A. L'erreur minimisée se développe comme suit :

$$\begin{aligned} \|\mathsf{A}\mathbf{h}\|^2 &= \cdots + (\mathbf{l}'^{\mathsf{T}}_{i'j}\mathsf{P}'_{i'}\mathsf{H}\mathbf{Q}_{jk})^2 + \cdots \\ &= \cdots + d^2_{A_PH}(\mathsf{P}'_{i'}\mathsf{H}\mathbf{Q}_{jk}, \mathbf{l}'_{i'j}) + \cdots . \end{aligned} \tag{8.7}$$

Chaque terme de cette erreur est une distance algébrique entre une droite image observée et un point prédit par transfert puis reprojection. La valeur de cette distance dépend de l'échelle libre des coordonnées homogènes $\mathbf{Q}_{jk}$ et $\mathsf{P}'_{i'}$.

On observe très clairement la forme "duale" de la fonction de coût, par rapport au maximum de vraisemblance, équation (8.2), basée sur l'erreur de reprojection, équation (8.1). L'erreur est en effet exprimée entre des points reprojetés et des droites observées et non entre des droites reprojetées et les points mesurés. La minimisation d'une telle fonction de coût peut donner de bons résultats, malgrés cette différence de forme.

Contraintes basées image - méthode quasi-linéaire QLIN_2D. Les deux méthodes précédentes sont basées sur la minimisation de distances algébriques. Les fonctions de coût correspondantes n'ont donc pas de sens physique. La quasi-linéarisation est une technique permettant de compenser le biais entre une distance algébrique et la distance Euclidienne correspondante, si cette dernière existe, voir §2.6.2, chapitre 2. La méthode LIN_3D, basée sur une distance 3D, ne peut pas être quasi-linéarisée. En effet, il n'existe pas de distance Euclidienne en espace projectif. Par contre, la méthode LIN_2D peut être quasi-linéarisée, afin que la distance Euclidienne entre les points prédits et les droite observées soit minimisée, au lieu d'une distance algébrique.

Les coordonnées homogènes $\mathbf{l}'_{i'j}$ des droites observées peuvent être normalisées telles que le biais entre les distances algébrique et Euclidienne ne dépende plus que du facteur d'échelle des points prédits. Les facteurs de biais permettent de compenser itérativement ce biais. Par analogie avec l'équation (8.7), la fonction d'erreur minimisée par cet algorithme est :

$$\cdots + d^2_{PH}(\mathsf{P}'_{i'}\mathsf{H}\mathbf{Q}_{jk}, \mathbf{l}'_{i'j}) + \cdots ,$$

où d_{PH} est la distance Euclidienne entre un point et une droite, définie par l'équation (2.14). Notons que cette méthode hérite de la forme de la fonction de coût "duale" de la méthode linéaire.

8.5.1.2 Méthodes basées sur la matrice d'homographie pour droites 3D

La matrice d'homographie pour droites 3D est une matrice (6×6) permettant d'appliquer une homographie directement aux coordonnées de Plücker de droites 3D. Nous proposons quatre méthodes basées sur la matrice d'homographie pour droites 3D. La première méthode est basée sur des contraintes 3D et les trois autres sur des contraintes basées image.

Nous rappelons la forme de la matrice d'homographie pour droites 3D, notée $\tilde{\mathsf{H}}$. Si l'on partitionne la matrice d'homographie usuelle comme :

$$\mathsf{H}_{(4\times 4)} \quad \sim \quad \begin{pmatrix} \bar{\mathsf{H}} & \mathbf{h}_1 \\ \mathbf{h}_2^{\mathsf{T}} & \bar{h} \end{pmatrix}, \tag{8.8}$$

alors :

$$\tilde{\mathsf{H}}_{(6\times 6)} \sim \begin{pmatrix} \bar{\mathsf{H}}^{\star} & [\mathbf{h}_1]_{\wedge}\bar{\mathsf{H}} \\ -\bar{\mathsf{H}}[\mathbf{h}_2]_{\wedge} & h\bar{\mathsf{H}} - \mathbf{h}_1\mathbf{h}_2^{\mathsf{T}} \end{pmatrix}. \tag{8.9}$$

Nous renvoyons au chapitre précédent pour l'étude de la structure de cette matrice. L'idée clef des méthodes basées sur la matrice d'homographie pour droites 3D est de représenter les droites en coordonnées de Plücker et d'interpréter ces coordonnées comme des points de l'espace projectif $\mathbb{P}^5$. Comme mentionné en §2.12, chapitre 2, ces points sont sur la quadrique de Klein. La matrice d'homographie pour droites 3D est interprétée comme une homographie de $\mathbb{P}^5$ et peut être estimée de manière linéaire avec 7 correspondances de droites 3D. Les méthodes basées sur la matrice d'homographie pour droites 3D ne sont donc pas minimales. Bien sûr, l'estimation linéaire est effectuée en négligeant les contraintes de consistance, données par les équations (7.7), (7.8) et (7.10), c'est à dire la forme spécifique de la matrice d'homographie pour droites 3D. Nos méthodes sont donc découpées en deux phases : *estimation* et *extraction*. Lors de la phase d'estimation, une matrice (6×6) B, représentant une homographie de $\mathbb{P}^5$, est estimée de manière linéaire. Lors de la phase d'extraction, une matrice d'homographie usuelle H est extraite de B. Un algorithme d'extraction est proposé au chapitre 7, §7.4.1.

La phase d'extraction introduit un biais dans l'estimation. Cependant, nous montrons ci-dessous que la forme de la fonction de coût minimisée par la méthode 3DLMM_LIN_2D_2 (voir ci-dessous), lors de la phase d'estimation, est plus justifiée que celle considérée par les méthodes décrites précédemment, basées sur la matrice d'homographie usuelle, voir §8.5.1.1. Nous décrivons ci-dessous les phases d'estimation des différentes méthodes.

Contraintes 3D - méthode 3DLMM_LIN_3D. Soit $\boldsymbol{L}_j \leftrightarrow \boldsymbol{L}'_j$ une correspondance de droites et $\mathbf{L}_j$ et $\mathbf{L}'_j$ leurs coordonnées de Plücker. Nous proposons une contrainte 3D illustrée sur la figure 8.4. Soit $\tilde{\mathsf{H}}$ la matrice

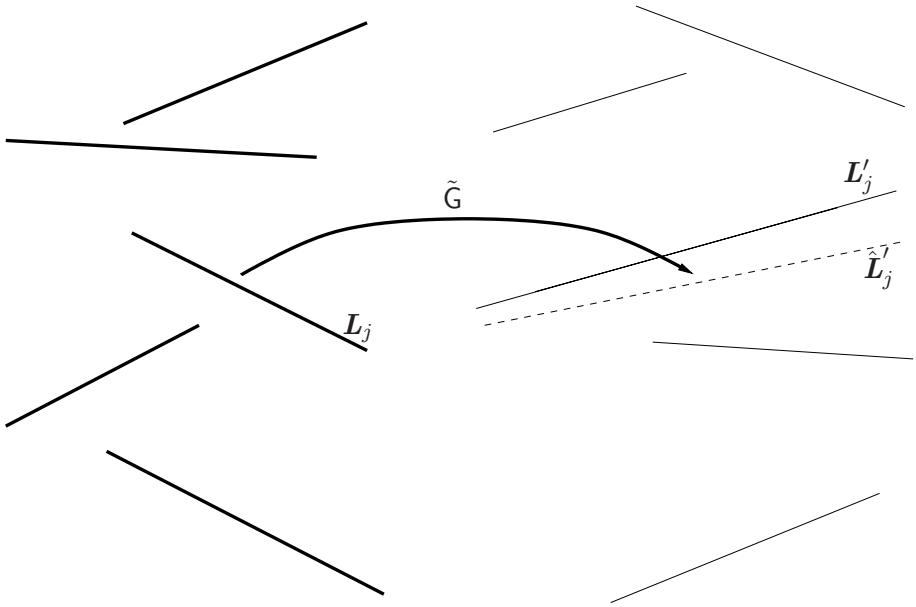


FIG. 8.4 – Le principe de la formulation de contraintes 3D linéaires sur les coefficients d'une matrice de mouvement pour droites 3D $\tilde{\mathsf{G}}$: les droites sont représentées par des coordonnées de Plücker. L'erreur est mesurée entre les droites $\boldsymbol{L}_j$ de la première base transférées vers la deuxième base en $\hat{\boldsymbol{L}}'_j$ et les droites de la deuxième base $\boldsymbol{L}'_j$.

d'homographie pour droites 3D reliant les deux reconstructions. Cette matrice permet de transférer les droites exprimées en coordonnées de Plücker. Soient $\hat{\mathbf{L}}'_j \sim \tilde{\mathsf{H}}\mathbf{L}_j$ les coordonnées des droites transférées, nous avons la relation :

$$\hat{\mathbf{L}}'_j \;\sim\; \mathbf{L}'_j.$$

L'égalité à un facteur près de cette équation induit des contraintes linéaires sur les coefficients de la matrice $\tilde{\mathsf{H}}$. Chaque correspondance induit 5 contraintes indépendantes. Ces contraintes sont liées à la mesure de distance algébrique d_{A_VV}, entre vecteurs de coordonnées homogènes, voir la section 2.5.2 du chapitre 2. Cette distance est donnée par l'équation (2.15). Elle provient du fait que $\forall c \in \{1\dots6\}$, $\forall t \in \{1\dots6\}$:

$$L'_{j,c}\hat{L}'_{j,t} - L'_{j,t}\hat{L}'_{j,c} \;=\; 0, \tag{8.10}$$

avec $\mathbf{L}'^{\mathsf{T}}_j = (L'_{j,1}\ L'_{j,2}\ L'_{j,3}\ L'_{j,4}\ L'_{j,5}\ L'_{j,6})$ et $\hat{\mathbf{L}}'^{\mathsf{T}}_j = (\hat{L}'_{j,1}\ \hat{L}'_{j,2}\ \hat{L}'_{j,3}\ \hat{L}'_{j,4}\ \hat{L}'_{j,5}\ \hat{L}'_{j,6})$. Des 15 équations possibles, seules 5 sont indépendantes, mais on ne peut pas en sélectionner 5 a priori (voir §2.5.2, chapitre 2).

Pour formuler le système linéaire induit sur les coefficients de $\tilde{\mathsf{H}}$, on remplace dans l'équation (8.10), l'expression $\hat{L}'_{j,c} = (\tilde{\mathsf{H}}\mathbf{L}_j)_c = \mathbf{h}_c^{\mathsf{T}}\mathbf{L}_j = \mathbf{L}_j^{\mathsf{T}}\mathbf{h}_c$, où $\mathbf{h}_c$ est la $c^{\text{ème}}$ ligne de la matrice $\tilde{\mathsf{H}}$. Les contraintes se reécrivent :

$$L'_{j,c}\mathbf{L}_j^{\mathsf{T}}\mathbf{h}_t - L'_{j,t}\mathbf{L}_j^{\mathsf{T}}\mathbf{h}_c \;=\; 0.$$

Notons $\tilde{\mathbf{h}} = \text{vect}(\tilde{\mathsf{H}})$ la vectorisation par ligne de $\tilde{\mathsf{H}}$. Si m correspondances sont utilisées, pour $c \in \{1\dots6\}$, $t \in \{1\dots6\}$ et $c > t$, un système de taille $(15m \times 36)$ est construit, donné sous forme matricielle par :

$$\mathsf{A}_{(15m\times36)}\tilde{\mathbf{h}}_{(36\times1)} \;=\; \mathbf{0}_{(15m\times1)} \quad \text{avec} \quad \mathsf{A}^{\mathsf{T}} \;=\; \begin{pmatrix}\mathsf{A}_1^{\mathsf{T}} & \cdots & \mathsf{A}_m^{\mathsf{T}}\end{pmatrix},$$

où :

$$\mathsf{A}_{j\,(15\times36)} \;=\; \begin{pmatrix}
L'_{j,2}\mathbf{L}_j^{\mathsf{T}} & -L'_{j,1}\mathbf{L}_j^{\mathsf{T}} & & & & \\
L'_{j,3}\mathbf{L}_j^{\mathsf{T}} & & -L'_{j,1}\mathbf{L}_j^{\mathsf{T}} & & & \\
L'_{j,4}\mathbf{L}_j^{\mathsf{T}} & & & -L'_{j,1}\mathbf{L}_j^{\mathsf{T}} & & \\
L'_{j,5}\mathbf{L}_j^{\mathsf{T}} & & & & -L'_{j,1}\mathbf{L}_j^{\mathsf{T}} & \\
L'_{j,6}\mathbf{L}_j^{\mathsf{T}} & & & & & -L'_{j,1}\mathbf{L}_j^{\mathsf{T}} \\
 & L'_{j,3}\mathbf{L}_j^{\mathsf{T}} & -L'_{j,2}\mathbf{L}_j^{\mathsf{T}} & & & \\
 & L'_{j,4}\mathbf{L}_j^{\mathsf{T}} & & -L'_{j,2}\mathbf{L}_j^{\mathsf{T}} & & \\
 & L'_{j,5}\mathbf{L}_j^{\mathsf{T}} & & & -L'_{j,2}\mathbf{L}_j^{\mathsf{T}} & \\
 & L'_{j,6}\mathbf{L}_j^{\mathsf{T}} & & & & -L'_{j,2}\mathbf{L}_j^{\mathsf{T}} \\
 & & L'_{j,4}\mathbf{L}_j^{\mathsf{T}} & -L'_{j,3}\mathbf{L}_j^{\mathsf{T}} & & \\
 & & L'_{j,5}\mathbf{L}_j^{\mathsf{T}} & & -L'_{j,3}\mathbf{L}_j^{\mathsf{T}} & \\
 & & L'_{j,6}\mathbf{L}_j^{\mathsf{T}} & & & -L'_{j,3}\mathbf{L}_j^{\mathsf{T}} \\
 & & & L'_{j,5}\mathbf{L}_j^{\mathsf{T}} & -L'_{j,4}\mathbf{L}_j^{\mathsf{T}} & \\
 & & & L'_{j,6}\mathbf{L}_j^{\mathsf{T}} & & -L'_{j,4}\mathbf{L}_j^{\mathsf{T}} \\
 & & & & L'_{j,6}\mathbf{L}_j^{\mathsf{T}} & -L'_{j,5}\mathbf{L}_j^{\mathsf{T}}
\end{pmatrix}.$$

En pratique, dû au bruit dans les mesures, le rang de la matrice A est plein. Nous minimisons l'erreur $\|\mathsf{A}\tilde{\mathbf{h}}\|^2$ sous la contrainte $\|\tilde{\mathbf{h}}\|^2 = 1$. La solution pour $\tilde{\mathbf{h}}$ est alors donnée par le vecteur singulier associé à la plus petite valeur singulière de la matrice A. La fonction de coût se développe en :

$$\|\mathsf{A}\tilde{\mathbf{h}}\|^2 \;=\; \cdots + d^2_{A_VV}(\mathbf{L}'_j, \tilde{\mathsf{H}}\mathbf{L}_j) + \cdots.$$

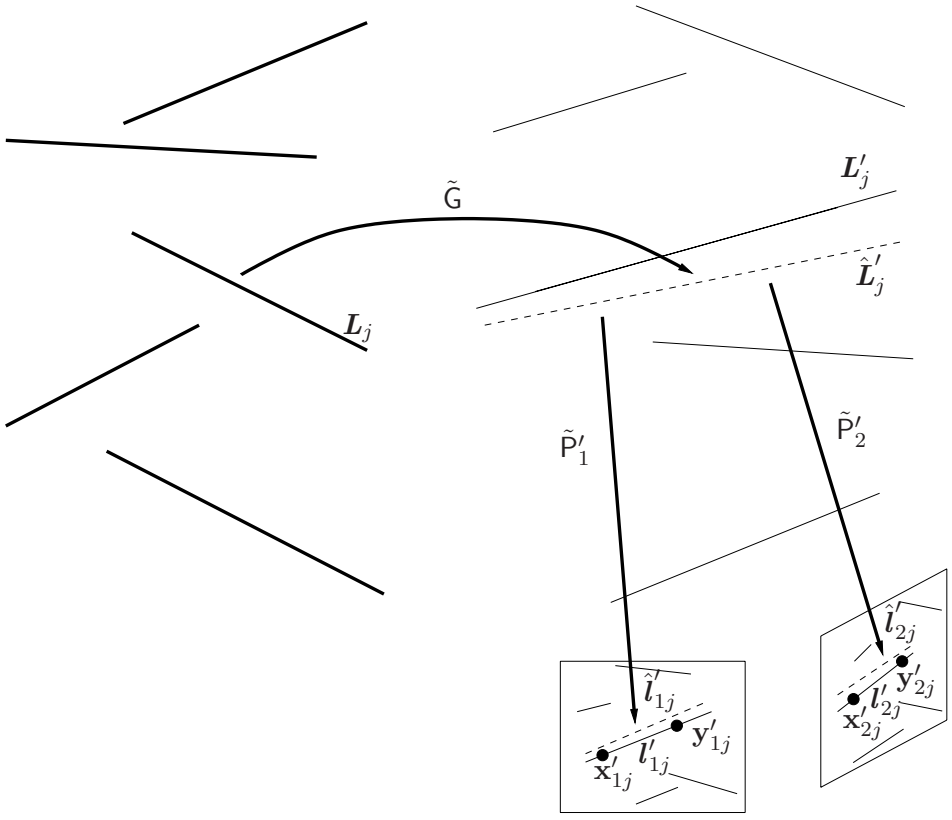


FIG. 8.5 – Le principe de la formulation de contraintes basées image, linéaires sur les coefficients d'une matrice de mouvement pour droites 3D $\tilde{\mathsf{G}}$: les droites de la première reconstruction sont représentées par des coordonnées de Plücker et celles de la deuxième reconstruction par des droites ou points image. L'erreur est mesurée entre les droites $\boldsymbol{L}_j$ transférées de la première vers la deuxième base et reprojetées dans le deuxième jeu d'images en $\hat{\boldsymbol{l}}'_{i'j}$ d'une part, et d'autre part, des points mesurés $\boldsymbol{x}'_{i'j}$ et $\boldsymbol{y}'_{i'j}$ ou des droites observées $\boldsymbol{l}'_{i'j}$ dans le deuxième jeu d'images.

Contraintes basées image 1 - méthode 3DLMM_LIN_2D_1. Nous suivons un principe similaire au cas 3D précédent, mais notre but est d'exprimer les contraintes en nous basant sur des entités image. L'idée est illustrée sur la figure 8.5. Considérons les coordonnées de Plücker de la droite transférée $\hat{\mathbf{L}}'_j \sim \tilde{\mathsf{H}}\mathbf{L}_j$. Notons $\mathsf{P}'_{i'}$ les matrices de projection associées aux caméras de la deuxième reconstruction. A partir de ces matrices, nous pouvons former les matrices de projection (3×6) pour coordonnées de Plücker $\tilde{\mathsf{P}}'_{i'}$ définies par l'équation (6.3). Cette méthode d'estimation est basée sur la comparaison directe des équations de droites dans les images. Reprojetons les droites transférées $\hat{\boldsymbol{L}}'_j$. Nous obtenons $\hat{\mathbf{l}}'_{i'j} \sim \tilde{\mathsf{P}}'_{i'}\hat{\mathbf{L}}'_j \sim \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j$. Nous utilisons la mesure de distance algébrique entre vecteurs homogènes (2.15) afin de comparer les droites prédites $\hat{\boldsymbol{l}}'_{i'j}$ et les droites mesurées $\boldsymbol{l}'_{i'j}$:

$$d^2_{A_VV}(\mathbf{l}'_{i'j}, \hat{\mathbf{l}}'_{i'j}) \quad = \quad \|\mathbf{l}'_{i'j} \wedge \hat{\mathbf{l}}'_{i'j}\|^2.$$

Ces contraintes permettent de construire un système linéaire pour les coefficients de $\tilde{\mathsf{H}}$. Chaque reprojection donne 3 équations, dont 2 seulement sont indépendantes. Si m correspondances de droites sont utilisées et que n' est le nombre d'images du second jeu, nous obtenons un système linéaire de taille $(3mn' \times 36)$.

Pour construire ce système, remplaçons l'expression de $\hat{\boldsymbol{l}}'_{i'j}$ dans l'équation ci-dessus. Nous obtenons :

$$d^2_{A_VV}(\mathbf{l}'_{i'j}, \hat{\mathbf{l}}'_{i'j}) \quad = \quad \|[\mathbf{l}'_{i'j}]_\wedge \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j\|^2,$$

que l'on peut reécrire, en notant $\tilde{\mathbf{h}} = \text{vect}(\tilde{\mathsf{H}})$, comme :

$$\mathsf{A}_{(3mn' \times 36)}\tilde{\mathbf{h}}_{(36\times 1)} \quad = \quad \mathbf{0}_{(3mn'\times 1)}, \quad \text{avec} \quad \mathsf{A} \quad = \quad \begin{pmatrix} \cdots \\ [\mathbf{l}'_{i'j}]_\wedge \tilde{\mathsf{P}}'_{i'} \text{diag}(\mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T}) \\ \cdots \end{pmatrix}.$$

En pratique, dû au bruit dans les mesures, le rang de la matrice A est plein. Nous minimisons l'erreur $\|\mathsf{A}\tilde{\mathbf{h}}\|^2$ sous la contrainte $\|\tilde{\mathbf{h}}\|^2 = 1$. La solution pour $\tilde{\mathbf{h}}$ est alors donnée par le vecteur singulier associé à la plus petite valeur singulière de la matrice A. La fonction de coût se développe en :

$$\|\mathsf{A}\tilde{\mathbf{h}}\|^2 \quad = \quad \cdots + d^2_{A_VV}(\mathbf{l}'_{i'j}, \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j) + \cdots . \tag{8.11}$$

Contraintes basées image 2 - méthode 3DLMM_LIN_2D_2. L'idée suivie est proche du cas précédent. La différence est qu'on ne se base pas sur une comparaison entre les droites observées et les droites prédites, mais entre les *points mesurés* et les droites prédites. Soient $\boldsymbol{x}'_{i'j}$ et $\boldsymbol{y}'_{i'j}$ des points mesurés sur la $j^{\text{ème}}$ droite observée dans la $i'^{\text{ème}}$ caméra et $\hat{\boldsymbol{l}}'_{i'j}$ la $j^{\text{ème}}$ droite transférée de la première base et reprojetée sur la $i'^{\text{ème}}$ image. L'appartenance des points à la droite s'écrit :

$$\mathbf{x}'^\mathsf{T}_{i'j}\hat{\mathbf{l}}'_{i'j} \quad = \quad \mathbf{y}'^\mathsf{T}_{i'j}\hat{\mathbf{l}}'_{i'j} \quad = \quad 0,$$

que l'on peut écrire, en remplaçant $\hat{\mathbf{l}}'_{i'j}$ par l'expression $\hat{\mathbf{l}}'_{i'j} \sim \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j$:

$$\mathbf{x}'^\mathsf{T}_{i'j}\tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j \quad = \quad \mathbf{y}'^\mathsf{T}_{i'j}\tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j \quad = \quad 0.$$

Chaque point fournit une contrainte. Si les droites image sont représentées par deux points dans chaque image, on obtient un système linéaire de taille $(2mn' \times 36)$, donné par :

$$\mathsf{A}_{(2mn' \times 36)}\tilde{\mathbf{h}}_{(36\times 1)} \quad = \quad \mathbf{0}_{(2mn'\times 1)} \quad \text{avec} \quad \mathsf{A} \quad = \quad \begin{pmatrix} \cdots \\ \mathbf{x}'^\mathsf{T}_{i'j}\tilde{\mathsf{P}}'_{i'} \text{diag}(\mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T}) \\ \mathbf{y}'^\mathsf{T}_{i'j}\tilde{\mathsf{P}}'_{i'} \text{diag}(\mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T},\ \mathbf{L}_j^\mathsf{T}) \\ \cdots \end{pmatrix}.$$

En pratique, dû au bruit dans les mesures, le rang de la matrice A est plein. Nous minimisons l'erreur $\|\mathsf{A}\tilde{\mathbf{h}}\|^2$ sous la contrainte $\|\tilde{\mathbf{h}}\|^2 = 1$. La solution pour $\tilde{\mathbf{h}}$ est alors donnée par le vecteur singulier associé à la plus petite valeur singulière de la matrice A. La fonction de coût se développe en :

$$\|\mathsf{A}\tilde{\mathbf{h}}\|^2 \quad = \quad \cdots + d^2_{A_PH}(\mathbf{x}'_{i'j}, \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j) + d^2_{A_PH}(\mathbf{y}'_{i'j}, \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j) + \cdots . \tag{8.12}$$

Contraintes basées image - méthode quasi-linéaire 3DLMM_QLIN_2D. Cette méthode est basée sur l'application du schéma de résolution quasi-linéaire donné en §2.6.2, chapitre 2, à la méthode 3DLMM_LIN_2D_2 du paragraphe précédent. Notons que c'est la seule des trois méthodes présentées ci-dessus pouvant être quasi-linéarisée, car c'est la seule dont la fonction de coût peut être exprimée avec une distance algébrique pour laquelle il existe une distance Euclidienne de même nature. Par exemple, la méthode 3DLMM_LIN_2D_1, basée sur la comparaison de droites image, ne peut pas être quasi-linéarisée car il n'existe pas de distance Euclidienne universelle entre deux droites image. La fonction de coût minimisée par 3DLMM_QLIN_2D est donnée en remplaçant la distance algébrique par la distance Euclidienne dans la fonction de coût de 3DLMM_LIN_2D_2, équation (8.12) :

$$\|\mathsf{A}\tilde{\mathbf{h}}\|^2 \quad = \quad \cdots + d^2_{PH}(\mathbf{x}'_{i'j}, \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j) + d^2_{PH}(\mathbf{y}'_{i'j}, \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j) + \cdots .$$

La repondération concerne les facteurs de biais $1/(\hat{l}^2_{i'j,1} + \hat{l}^2_{i'j,2})$. On remarque que la forme de cette fonction de coût est proche de celle de l'erreur de reprojection, équation (8.1), car elle est basée sur une distance Euclidienne entre les points mesurés et les droites prédites par le modèle.

8.5.2 Cas affine

Nous proposons des méthodes pour l'estimation d'une affinité à partir de correspondances de droites. Comme dans le cas homographique, ces méthodes sont divisées en deux catégories, basées sur la matrice d'affinité usuelle ou la matrice d'affinité pour droites 3D. La structure de la matrice d'affinité pour droites 3D est explicitement prise en compte lors de la phase d'estimation, contrairement au cas projectif, où une matrice (6×6) générale est estimée.

8.5.2.1 Méthodes basées sur la matrice d'affinité usuelle

Nous proposons trois méthodes, analogues à celles proposées en §8.5.1.1 pour l'estimation d'une homographie. Nous donnons l'idée de base de chaque méthode puis une formulation pratique, et enfin, nous analysons la fonction de coût induite.

Contraintes 3D - méthode AFF_LIN_3D. L'idée, illustrée sur la figure 8.2 et expliquée pour LIN_3D, est de représenter chaque droite $\boldsymbol{L}_j$ de la première reconstruction par des points $\boldsymbol{Q}_{jk}$ et la droite correspondante de la deuxième reconstruction par des plans $\boldsymbol{\Pi}'_{jl}$. Les contraintes exercées sur le mouvement par cette correspondance de droites sont formulées de manière linéaire sur les coefficients de la matrice d'affinité A par :

$$\boldsymbol{\Pi}'^{\mathsf{T}}_{jl}\mathsf{A}\mathbf{Q}_{jk} \quad = \quad 0.$$

On suppose que les $\mathbf{Q}^{\mathsf{T}}_k = (Q_{k,1} \;\; Q_{k,2} \;\; Q_{k,3} \;\; Q_{k,4})$ sont normalisés tels que $Q_{k,4} = 1$. Soient $\mathbf{a}_1$, $\mathbf{a}_2$ et $\mathbf{a}_3$ les trois premières lignes de A, les contraintes se reécrivent :

$$\Pi'_{jl,1}\mathbf{Q}^{\mathsf{T}}_{jk}\mathbf{a}_1 + \Pi'_{jl,2}\mathbf{Q}^{\mathsf{T}}_{jk}\mathbf{a}_2 + \Pi'_{jl,3}\mathbf{Q}^{\mathsf{T}}_{jk}\mathbf{a}_3 \quad = \quad -\Pi'_{jl,4}.$$

Chaque correspondance de droites donne $rs \geq 4$ équations, où $r \geq 2$ est le nombre de points et $s \geq 2$ le nombre de plans. Avec m correspondances, on forme un système linéaire non-homogène de taille[5] $(mrs \times 12)$, donné par :

$$\mathsf{C}_{(mrs\times 12)}\mathbf{a}_{(12\times 1)} \quad = \quad \mathbf{b}_{(mrs\times 1)},$$

avec

$$\mathsf{C} \quad = \quad \begin{pmatrix} \cdots & & \\ \Pi'_{jl,1}\mathbf{Q}^{\mathsf{T}}_{jk} & \Pi'_{jl,2}\mathbf{Q}^{\mathsf{T}}_{jk} & \Pi'_{jl,3}\mathbf{Q}^{\mathsf{T}}_{jk} \\ \cdots & & \end{pmatrix} \quad \text{et} \quad \mathbf{b} \quad = \quad \begin{pmatrix} \cdots \\ -\Pi'_{jl,4} \\ \cdots \end{pmatrix},$$

[5] Le nombre d'équations mrs suppose que le même nombre de points et de plans est utilisé pour chaque correspondance de droites. Cette hypothèse n'est utilisée qu'à des fins d'allègement des notations et n'est pas du tout essentielle au raisonnement proposé.

où $\mathbf{a}$ est la vectorisation par ligne des trois premières lignes de A :

$$\mathbf{a}^\mathsf{T} \;=\; (\;\mathbf{a}_1^\mathsf{T} \quad \mathbf{a}_2^\mathsf{T} \quad \mathbf{a}_3{}^\mathsf{T}\;).$$

La solution au sens des moindres carrés est donnée via la pseudo-inverse de C : $\mathbf{a} = \mathsf{C}^\dagger\mathbf{b}$, voir §2.6.3, chapitre 2. Cette méthode minimise le critère d'erreur suivant :

$$\begin{aligned}\|\mathsf{C}\mathbf{a}-\mathbf{b}\|^2 &= \cdots + ((\mathsf{A}\mathbf{Q}_{jk})^\mathsf{T}\mathbf{\Pi}'_{jl})^2 + \cdots \\ &= \cdots + d^2_{PH}(\mathsf{A}\mathbf{Q}_{jk}, \mathbf{\Pi}'_{jl}) + \cdots .\end{aligned}$$

Cette expression est similaire à celle obtenue pour la méthode LIN_3D d'estimation d'une homographie. Elle s'écrit avec la distance Euclidienne d_{PH}, si certaines contraintes de normalisation sont satisfaites sur les données : $Q_{jk,4} = 1$ et $\|\bar{\mathbf{\Pi}}_{jl}\|^2 = 1$. Cependant, il faut garder en mémoire que la distance Euclidienne n'a pas de sens en espace affine, et donc considérer la fonction de coût comme une somme de distances algébriques.

Contraintes basées image - méthode AFF_LIN_2D. L'idée est proche du cas 3D et similaire à celle utilisée pour la méthode LIN_2D d'estimation d'une homographie. Les points représentant les droites transférées sont reprojetés dans les images de la deuxième reconstruction. Ces reprojections sont comparées avec les droites observées dans ces images afin de formuler les contraintes sur le mouvement. Remplaçons l'homographie H par une affinité A dans l'expression algébrique de ces contraintes donnée par l'équation (8.5). En supposant $Q_{jk,4} = 1$, ces contraintes se reécrivent :

$$\mathbf{l}'^\mathsf{T}_{i'j}\mathbf{p}'_{i',1}\mathbf{Q}^\mathsf{T}_{jk}\mathbf{a}_1 + \mathbf{l}'^\mathsf{T}_{i'j}\mathbf{p}'_{i',2}\mathbf{Q}^\mathsf{T}_{jk}\mathbf{a}_2 + \mathbf{l}'^\mathsf{T}_{i'j}\mathbf{p}'_{i',3}\mathbf{Q}^\mathsf{T}_{jk}\mathbf{a}_3 \;=\; -\mathbf{l}'^\mathsf{T}_{i'j}\mathbf{p}'_{i',4},$$

où $\mathbf{p}'_{i',c}$ désigne la $c^{\text{ème}}$ colonne de $\mathsf{P}'_{i'}$. Chaque correspondance de droites donne $rn' \geq 4$ équations, où $r \geq 2$ est le nombre de points et $n' \geq 2$ le nombre d'images de la deuxième reconstruction. Avec m correspondances, on obtient un système linéaire non-homogène de taille[6] $(mrn' \times 12)$:

$$\mathsf{C}_{(mrn'\times 12)}\mathbf{a}_{(12\times 1)} \;=\; \mathbf{b}_{(mrn')},$$

avec :

$$\mathsf{C} \;=\; \begin{pmatrix} \cdots \\ \mathbf{l}'^\mathsf{T}_{i'j}\mathbf{p}'_{i',1}\mathbf{Q}^\mathsf{T}_{jk} & \mathbf{l}'^\mathsf{T}_{i'j}\mathbf{p}'_{i',2}\mathbf{Q}^\mathsf{T}_{jk} & \mathbf{l}'^\mathsf{T}_{i'j}\mathbf{p}'_{i',3}\mathbf{Q}^\mathsf{T}_{jk} \\ \cdots \end{pmatrix} \quad \text{et} \quad \mathbf{b} \;=\; \begin{pmatrix} \cdots \\ -\mathbf{l}'^\mathsf{T}_{i'j}\mathbf{p}'_{i',4} \\ \cdots \end{pmatrix}.$$

La solution au sens des moindres carrés est donnée via la pseudo-inverse de C : $\mathbf{a} = \mathsf{C}^\dagger\mathbf{b}$, voir §2.6.3, chapitre 2. L'erreur minimisée par cette méthode est la même que celle de la méthode LIN_2D_1, équation (8.11), dans le cas homographique :

$$\|\mathsf{C}\mathbf{a}-\mathbf{b}\|^2 \;=\; \cdots + d^2_{A_VV}(\mathbf{l}'_{i'j}, \tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{H}}\mathbf{L}_j) + \cdots .$$

La valeur de cette fonction de coût dépend de l'échelle libre des matrices $\mathsf{P}'_{i'}$.

Contraintes basées image - méthode quasi-linéaire AFF_QLIN_2D. Cette méthode est issue de la quasi-linéarisation de la méthode AFF_LIN_2D, selon la méthode présenté en §2.6.2, chapitre 2. Elle minimise la fonction de coût suivante, identique à celle de QLIN_2D :

$$\cdots + d^2_{PH}(\mathsf{P}'_{i'}\mathsf{H}\mathbf{Q}_{jk}, \mathbf{l}'_{i'j}) + \cdots .$$

[6] Comme dans le cas 3D, le nombre d'équations mrn' suppose que le même nombre de points et de droites images est utilisé pour représenter chaque correspondance de droites. Cette hypothèse n'est utilisée qu'à des fins d'allègement des notations et n'est pas du tout essentielle au raisonnement proposé.

8.5.2.2 Méthodes basées sur la matrice d'affinité pour droites 3D

La matrice d'affinité pour droites 3D, de taille (6×6), permet d'appliquer une affinité directement aux coordonnées de Plücker de droites 3D. Nous proposons deux méthodes basées sur la matrice d'affinité pour droites 3D. La première méthode est basée sur des contraintes 3D et les trois autres sur des contraintes basées image. Nous rappelons ci-dessous la forme de la matrice d'affinité pour droites 3D, étudiée en détails en §7.2.3, chapitre 7. Si la matrice d'affinité est partitionnée comme :

$$\mathsf{A} \sim \begin{pmatrix} & \bar{\mathsf{A}} & & \mathbf{t} \\ 0 & 0 & 0 & a \end{pmatrix},$$

avec $a \neq 0$, alors la matrice d'affinité pour droites 3D est donnée par :

$$\tilde{\mathsf{A}} \sim \begin{pmatrix} \bar{\mathsf{A}}^\star & [\mathbf{t}]_\wedge \bar{\mathsf{A}} \\ \mathsf{0}_{(3\times 3)} & a\bar{\mathsf{A}} \end{pmatrix}.$$

Une première idée pour l'estimation de l'affinité est d'utiliser la technique mise en œuvre dans le cas homographique, voir §8.5.1.2. En interprétant les coordonnées de Plücker comme des points de $\mathbb{P}^5$, on peut concevoir des méthodes basées sur les deux phases estimation et extraction. Cependant, la forme spécifique de la matrice d'affinité pour droites 3D, contrairement à celle, plus compliquée, de la matrice d'homographie pour droites 3D, peut être explicitement prise en compte lors de l'estimation basée sur des contraintes 3D, en découplant la phase d'estimation en deux. La phase d'extraction n'est donc plus nécessaire. Dans le cas des contraintes basées image, seule la contrainte de consistance triviale correspondant au bloque (3×3) de zéros, en bas à gauche de $\tilde{\mathsf{A}}$, peut être prise en compte.

Contraintes 3D - méthode 3DLMM_AFF_LIN_3D. Cette méthode est basée sur le même principe que la méthode 3DLMM_LIN_3D dans le cas homographique. Les équations sont déduites de l'égalité à un facteur près des vecteurs de coordonnées homogènes représentant les coordonnées de Plücker des correspondances de droites. Nous proposons une méthode basée sur deux phases : *estimation I* et *estimation II*. La phase estimation I calcule le bloque $a\bar{\mathsf{A}}$, en bas à droite de $\tilde{\mathsf{A}}$, et la phase estimation II en déduit les éléments restants : a et $\mathbf{t}$, en se basant sur les équations induites sur les trois premières lignes de $\tilde{\mathsf{A}}$.

Soient $\mathbf{L}_j^\mathsf{T} \sim (\ \mathbf{a}_j^\mathsf{T} \quad \mathbf{b}_j^\mathsf{T}\ )$ et $\mathbf{L}'^\mathsf{T}_j \sim (\ \mathbf{a}'^\mathsf{T}_j \quad \mathbf{b}'^\mathsf{T}_j\ )$ les coordonnées de Plücker de la $j^{\text{ème}}$ correspondance de droites. D'après la définition de la matrice d'affinité pour droites 3D, elles sont reliées par :

$$\mathbf{L}'_j \sim \tilde{\mathsf{A}}\mathbf{L}_j,$$

qui peut être reécrit comme :

$$\mathbf{a}'_j \sim \bar{\mathsf{A}}^\star \mathbf{a}_j + [\mathbf{t}]_\wedge \bar{\mathsf{A}}\mathbf{b}_j \tag{8.13}$$
$$\mathbf{b}'_j \sim a\bar{\mathsf{A}}\mathbf{b}_j. \tag{8.14}$$

Lors de cette reécriture, de l'information est perdue, car les deux égalités à un facteur près obtenues ne modélisent pas explicitement le fait que les deux facteurs sont identiques. L'introduction explicite de tels facteurs ne permet pas de concevoir un estimateur linéaire dans le cas affine, mais se révèle utile dans le cas de l'estimation d'une similitude ou d'un mouvement rigide, voir la section suivante.

La phase estimation I utilise l'équation (8.14). La matrice $\bar{\mathsf{A}}$ ne peut être estimée qu'à un facteur près, c'est à dire que le facteur a ne peut être calculée. Pour mettre ceci en évidence, nous notons $\bar{\mathsf{M}} = a\bar{\mathsf{A}}$. Le facteur a ne sera déterminé qu'indirectement dans la phase estimation II. L'équation (8.14) se reécrit :

$$\mathbf{b}'_j \sim \bar{\mathsf{M}}\mathbf{b}_j.$$

La matrice $\bar{\mathsf{M}}$ peut être obtenue en minimisant un critère au sens des moindres carrés. On peut construire un système linéaire homogène de taille $(3m \times 9)$ permettant d'estimer $\bar{\mathsf{M}}$ basé sur le fait que $(\mathbf{b}'_j \sim \mathsf{M}\mathbf{b}_j) \Rightarrow ([\mathbf{b}'_j]_\wedge \mathsf{M}\mathbf{b}_j = \mathbf{0})$. On remarque que cette phase est algébriquement similaire à l'estimation d'une homographie 2D. Nous renvoyons au chapitre 2, §2.6.1 pour des détails et la résolution du système linéaire. D'un point de vue géométrique, ce raisonnement revient à n'utiliser que la partie directionnelle des droites pour estimer $\bar{\mathsf{M}}$, qui est lié à l'homographie à l'infini entre deux vues particulières des deux jeux de caméras, voir l'interprétation géométrique de la section 7.3.2 du chapitre 7.

La phase estimation II utilise l'équation (8.13). En remplaçant $\bar{\mathsf{A}} = \mu\bar{\mathsf{M}}$ dans cette équation, avec $\mu = 1/a$, et en notant que d'après l'équation (2.5), $(\mu\bar{\mathsf{M}})^\star = \mu^2\bar{\mathsf{M}}^\star$, on obtient

$$\mathbf{a}'_j \sim \mu^2\bar{\mathsf{M}}^\star\mathbf{a}_j + \mu[\mathbf{t}]_\wedge\bar{\mathsf{M}}\mathbf{b}_j \sim \mu\bar{\mathsf{M}}^\star\mathbf{a}_j + [\mathbf{t}]_\wedge\bar{\mathsf{M}}\mathbf{b}_j.$$

Cette équation implique :

$$[\mathbf{a}'_j]_\wedge \left(\mu\bar{\mathsf{M}}^\star\mathbf{a}_j + [\mathbf{t}]_\wedge\bar{\mathsf{M}}\mathbf{b}_j\right) = \mathbf{0}.$$

On construit alors un système linéaire homogène pour les inconnues, μ et $\mathbf{t}$:

$$\underset{(3\times4)}{\begin{pmatrix} [\mathbf{a}'_j]_\wedge\bar{\mathsf{M}}\mathbf{a}_j & -[\mathbf{a}'_j]_\wedge[\bar{\mathsf{M}}\mathbf{b}_j]_\wedge \end{pmatrix}} \underset{(4\times1)}{\begin{pmatrix} \mu \\ \mathbf{t} \end{pmatrix}} = \mathbf{0}_{(3\times1)}.$$

Ce système peut être résolu au sens des moindres carrés, voir la section 2.4.1, chapitre 2. Cette méthode n'est pas minimale. Alors que 3 correspondances de droites définissent une affinité, 4 sont requises ici. La phase estimation I est similaire à l'estimation d'une homographie 2D à partir de correspondances de points, voir §2.6.1.1, chapitre 2, où chaque correspondance de droites joue le rôle d'une correspondance de points. Elle nécessite donc 4 correspondances de droites. Cette méthode minimise successivement deux fonctions de coût, n'ayant pas de sens physique.

Contraintes basées image. La prise en compte de la forme spécifique de la matrice d'affinité pour droites 3D, en utilisant des contraintes basées image est plus difficile que dans le cas précédent. Les contraintes sont du type illustré sur la figure 8.5, page 191, et utilisées pour les méthodes 3DLMM_LIN_2D_*. Elles décrivent le fait que les points mesurés $\boldsymbol{x}_{ij}$ et $\boldsymbol{y}_{ij}$ doivent être sur la droite prédite $\hat{\boldsymbol{l}}_{ij}$. Cette droite est issue de la première reconstruction, elle est transférée dans la deuxième base à l'aide de $\tilde{\mathsf{A}}$, puis reprojetée dans le deuxième jeu d'images, par les matrices de projection $(3 \times 6)\,\tilde{\mathsf{P}}'_{i'}$. On obtient donc des équations de la forme :

$$\mathbf{x}_{i'j}^\mathsf{T}\tilde{\mathsf{P}}'_{i'}\tilde{\mathsf{A}}\mathbf{L}_j = 0.$$

En remplaçant $\tilde{\mathsf{A}}$ par son expression (7.12) et en écrivant $\tilde{\mathsf{P}}'_{i'} \sim \begin{pmatrix} \bar{\mathsf{P}}_{i',1} & \bar{\mathsf{P}}_{i',2} \end{pmatrix}$ et $\mathbf{L}_j^\mathsf{T} \sim \begin{pmatrix} \mathbf{a}_j^\mathsf{T} & \mathbf{b}_j^\mathsf{T} \end{pmatrix}$, on obtient :

$$\mathbf{x}_{i'j}^\mathsf{T}\bar{\mathsf{P}}_{i',1}\bar{\mathsf{A}}^\star\mathbf{a}_j + \mathbf{x}_{i'j}^\mathsf{T}\bar{\mathsf{P}}_{i',1}[\mathbf{t}]_\wedge\bar{\mathsf{A}}\mathbf{b}_j + \mathbf{x}_{i'j}^\mathsf{T}\bar{\mathsf{P}}_{i',2}a\bar{\mathsf{A}}\mathbf{b}_j = 0.$$

Cette équation est bilinéaire car elle implique des produits entre différents éléments de la matrice d'affinité A, et l'estimation ne peut pas être effectuée en deux phases comme dans le cas 3D. Cette équation peut toutefois être utilisée pour estimer des bloques de $\tilde{\mathsf{A}}$, en ignorant les contraintes de consistance (seule la contrainte de consistance triviale $\tilde{\mathsf{A}}_{21} = \mathsf{0}_{(3\times3)}$ est alors prise en compte). Pour ce faire, on pose $\tilde{\mathsf{A}}_{11} = \bar{\mathsf{A}}^\star\mathbf{a}$, $\tilde{\mathsf{A}}_{12} = [\mathbf{t}]_\wedge\mathsf{A}$ et $\tilde{\mathsf{A}}_{22} = a\mathsf{A}$. L'affinité est extraite de ces bloques avec l'algorithme d'extraction donné en §7.4.2, chapitre 7.

Comme dans le cas homographique, on peut utiliser des contraintes entre les droites reprojetées et les droites observées (voir méthode 3DLMM_LIN_2D_1), ou entre les droites reprojetées et les points mesurés (voir méthode 3DLMM_LIN_2D_2), et quasi-linéariser cette dernière méthode (voir méthode 3DLMM_QLIN_2D).

8.5.3 Cas métrique et Euclidien

Ce cas est différent des cas affine et homographique, dans le sens où les matrices de similitude et de mouvement rigide recherchées contiennent une matrice de rotation 3D. En conséquence, des contraintes non-linéaires doivent être satisfaites, concernant l'orthonormalité de la matrice de rotation. De telle contraintes sont difficiles à manipuler, et rendent l'estimation linéaire indirecte. Nous proposons des méthodes basées sur le calcul préliminaire d'une affinité, c'est à dire ignorant la plupart des contraintes d'orthonormalité, puis l'extraction du mouvement.

8.5.3.1 Méthodes basées sur le calcul préliminaire d'une affinité

Plusieurs approches sont possibles : calculer l'affinité puis en extraire le mouvement en forçant les contraintes d'orthonormalité, ou remplacer, dans les algorithmes de calcul d'affinité basés sur la matrice d'affinité pour droites 3D, l'étape finale d'extraction de l'affinité, par l'extraction d'une similitude ou d'un mouvement rigide.

Dans le premier cas, lorsqu'une affinité est estimée, le bloque (3×3) supérieur $\bar{\mathsf{A}}$ doit être corrigé pour être une matrice de rotation (à l'échelle dans le cas de la similitude). Une telle correction peut-être faite de la manière suivante. Soit $\bar{\mathsf{A}} = \mathsf{U}\,\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)\,\mathsf{V}^\mathsf{T}$ une décomposition en valeurs singulières de $\bar{\mathsf{A}}$. Dans le cas de la similitude, le facteur d'échelle est donné par $s = (\sigma_1 + \sigma_2 + \sigma_3)/3$. La matrice de rotation est donnée par $\mathsf{R} = \det(\mathsf{U})\det(\mathsf{V})\mathsf{U}\mathsf{V}^\mathsf{T}$ (le facteur $\det(\mathsf{U})\det(\mathsf{V})$ garantit que la matrice R a un déterminant positif). Cette correction garantit que $s\mathsf{R}$ est la matrice de rotation à l'échelle la plus proche de $\bar{\mathsf{A}}$, au sens de la norme de Frobenius.

Dans le deuxième cas, lorsqu'une matrice d'affinité pour droites 3D est estimée par les phases estimation puis extraction, la phase d'extraction d'une affinité est remplacée par l'extraction d'une similitude ou d'un mouvement rigide, par exemple par les algorithmes donnés en §§7.4.3 et 7.4.4, chapitre 7.

Notons que ces méthodes ne peuvent être minimales dans aucun cas, car trois correspondances de droites sont nécessaires pour le calcul d'une affinité, alors que deux correspondances définissent une similitude ou un mouvement rigide. La fonction de coût minimisée par cette méthode dépend de la méthode sous-jacente utilisée pour estimer l'affinité.

8.5.3.2 Méthodes basées sur les matrices de similitude et de mouvement rigide pour droites 3D

Contraintes 3D - méthode 3DLMM_SIM_LIN_3D. L'idée suivie est similaire au cas affine, méthode 3DLMM_AFF_LIN_3D. Elle consiste à découper l'estimation en deux phases. La phase estimation I calcule la rotation, et la phase estimation II, la translation, et le facteur d'échelle dans le cas de la similitude. En suivant la démarche utilisée dans le cas affine, on est conduit aux équations suivantes, où l'égalité à un facteur près est modélisée explicitement par un ensemble de facteurs inconnus $\lambda_j \in \mathbb{R}^*$:

$$\lambda_j \mathbf{a}'_j = s\mathsf{R}\mathbf{a}_j + [\mathbf{t}]_\wedge \mathsf{R}\mathbf{b}_j \tag{8.15}$$

$$\lambda_j \mathbf{b}'_j = \mathsf{R}\mathbf{b}_j. \tag{8.16}$$

La phase estimation I utilise l'équation (8.16) pour calculer R et les facteurs λ_j. Cette équation correspond à la partie directionnelle des droites, encapsulée dans la partie inférieure des coordonnées de Plücker, ce qui explique pourquoi cette équation est indépendante de l'échelle et de la translation. Le calcul des facteurs λ_j peut être effectué comme suit. En prenant la norme de chaque côté de l'équation (8.16), on obtient $\|\lambda_j \mathbf{b}'_j\| = \|\mathsf{R}\mathbf{b}_j\|$. Comme R est une matrice de rotation, $\|\mathsf{R}\mathbf{b}_j\| = \|\mathbf{b}_j\|$, et on obtient la valeur absolue de λ_j comme :

$$|\lambda_j| = \frac{\|\mathbf{b}_j\|}{\|\mathbf{b}'_j\|}.$$

Notons que $\|\mathbf{b}_j\|$ ou $\|\mathbf{b}'_j\|$ est nulle si et seulement si la droite correspondante est sur le plan à l'infini. Comme seule la valeur absolue de λ_j est déterminée, les vecteurs de coordonnées homogènes $\mathbf{L}_j$ et $\mathbf{L}'_j$ doivent être normalisés de telle sorte que les vecteurs représentant les directions $\mathbf{b}_j$ et $\mathbf{b}'_j$ pointent vers le même côté de la

sphère unité. La rotation peut ensuite être calculée en minimisant le critère suivant, après normalisation des $\mathbf{L}'_j$ par les λ_j :

$$\sum_{j=1}^{m} d_{PP}^2(\mathbf{b}'_j, \mathsf{R}\mathbf{b}_j),$$

où d_{PP} est la distance Euclidienne entre deux points, donnée par l'équation (2.13). Pour ce faire, nous suivons la méthode proposée par Horn et al. [102]. La solution minimisant le critère ci-dessus est donnée par la matrice orthonormale la plus proche de la matrice $\mathsf{N}_{(3\times3)} = \sum_{j=1}^{m} \mathbf{b}'_j \mathbf{b}_j^\mathsf{T}$. Cette solution est donnée par $\mathsf{R} = \det(\mathsf{U})\det(\mathsf{V})\mathsf{U}\mathsf{V}^\mathsf{T}$, où $\mathsf{N} = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$ est une décomposition en valeurs singulières.

La phase estimation II utilise l'équation (8.15) pour calculer la translation $\mathbf{t}$ et éventuellement l'échelle s. En suivant le même raisonnement que dans le cas affine, cette équation se reécrit en :

$$\underset{(3\times4)}{\begin{pmatrix} \mathsf{R}\mathbf{a}_j & -[\mathsf{R}\mathbf{b}_j]_\wedge \end{pmatrix}} \underset{(4\times1)}{\begin{pmatrix} s \\ \mathbf{t} \end{pmatrix}} = \underset{(3\times1)}{\begin{pmatrix} \lambda_j \mathbf{a}'_j \end{pmatrix}}.$$

En rassemblant les équations pour les différentes correspondances, on obtient un système linéaire non-homogène, qui peut être résolu au sens des moindres carrés. Dans le cas d'un mouvement rigide, on pose $s = 1$, ce qui simplifie le système.

Nous verrons en §8.7.2.1 que cette méthode donne de mauvais résultats, comparé à une méthode basée sur l'affinité, avec un nombre de correspondances de droites donnant des contraintes redondantes sur le mouvement. En effet, la fonction de coût minimisée n'a pas de sens physique. Son intérêt principal est donc d'être minimale, c'est à dire qu'elle permet de calculer un mouvement à partir du minimum de 2 correspondances de droites. Elle peut donc être utilisée, par exemple, dans le cadre d'un estimateur robuste basé sur un échantillonnage aléatoire, tel RANSAC [64].

Contraintes basées image. L'utilisation des contraintes basées image se solde, comme dans le cas affine, par des équations non-linéaires en les coefficients du mouvement recherché. Il faut donc "linéariser" ces équations en introduisant trois bloques intermédiaires, $\tilde{\mathsf{D}}_{11}$, $\tilde{\mathsf{D}}_{12}$, $\tilde{\mathsf{D}}_{22}$, car $\tilde{\mathsf{D}}_{21} = 0$. La similitude ou le mouvement rigide est alors extrait de ces bloques par les algorithmes proposés en §§7.4.3 et 7.4.4, chapitre 7. La linéarisation revient à calculer une affinité. La fonction de coût dépend de la méthode utilisée pour cela.

8.6 Méthodes non-linéaires

Nous décrivons deux méthodes non-linéaires pour l'estimation du mouvement 3D. Notons que les méthodes quasi-linéaires sont décrites dans la section précédente. Comme le maximum de vraisemblance, décrit en §8.4, ces méthodes ne dépendent pas intrinsèquement du type de mouvement estimé. Elles minimisent un critère d'erreur indépendant du type de mouvement, qui est déterminé par la paramétrisation employée. Nous utilisons l'algorithme de Levenberg-Marquardt pour conduire l'optimisation non-linéaire, comme dans le cas du maximum de vraisemblance, voir la section 8.4.2 pour plus de détails. Les droites 3D n'étant pas modifiées par ces estimateurs, nous les représentons par leurs coordonnées de Plücker.

Erreur de transfert non-symétrique - méthode NLIN_1. Cette méthode minimise une erreur de transfert exprimée dans le deuxième jeu d'images, sur les paramètres du mouvement. Cette fonction de coût correspond à celle minimisée par la méthode 3DLMM_QLIN_2D. La différence entre les deux méthodes est la technique d'optimisation non-linéaire employée : la méthode 3DLMM_QLIN_2D est basée sur la repondération itérative d'un système linéaire, elle est simple à implémenter, mais estime une matrice (6×6) et nécessite l'extraction du mouvement à chaque étape. La méthode NLIN_1 présentée dans cette section est basée sur un algorithme plus général d'optimisation non-linéaire. L'optimisation est conduite directement sur les paramètres du mouvement. La phase d'extraction n'est donc pas nécessaire.

L'erreur minimisée est donnée par :

$$\begin{aligned}\mathcal{L}(\hat{\mathcal{S}}', \mathcal{M}') &= \sum_{j=1}^{m} \mathcal{L}(\hat{\boldsymbol{L}}'_j, \mathcal{M}') \\ &= \sum_{j=1}^{m}\sum_{i'=1}^{n'} \left(d^2_{PH}(\mathbf{x}_{i'j}, \hat{\mathbf{l}}_{i'j}) + d^2_{PH}(\mathbf{y}_{i'j}, \hat{\mathbf{l}}_{i'j}) \right),\end{aligned}$$

où $\hat{\mathcal{S}}'$ désigne les droites de la première reconstruction, exprimées dans la deuxième base grâce au mouvement estimé et $\mathcal{L}$ est l'erreur de reprojection donnée par l'équation (8.1). La forme de cette erreur de transfert est proche de celle de l'erreur de reprojection.

Erreur de transfert symétrique - méthode NLIN_2. La méthode présentée ci-dessus est basée sur la minimisation de l'erreur de transfert, qui est un critère ayant du sens, et donne des résultats acceptables. Cependant, elle ne donne pas un rôle symétrique aux deux jeux d'images et de droites, car l'erreur est mesurée seulement dans le deuxième jeu d'images. La méthode NLIN_2, proposée dans cette section, est basée sur la minisation d'une erreur dans les deux jeux d'images, sur les paramètres du mouvement. Cette erreur s'exprime par :

$$\mathcal{L}(\hat{\mathcal{S}}', \mathcal{M}') + \mathcal{L}(\hat{\mathcal{S}}, \mathcal{M}) = \sum_{j=1}^{m} \left(\mathcal{L}(\hat{\boldsymbol{L}}'_j, \mathcal{M}') + \mathcal{L}(\hat{\boldsymbol{L}}_j, \mathcal{M}) \right)$$

$$= \sum_{j=1}^{m} \left(\sum_{i'=1}^{n'} \left(d^2_{PH}(\mathbf{x}'_{i'j}, \hat{\mathbf{l}}'_{i'j}) + d^2_{PH}(\mathbf{y}'_{i'j}, \hat{\mathbf{l}}'_{i'j}) \right) + \sum_{i=1}^{n} \left(d^2_{PH}(\mathbf{x}_{ij}, \hat{\mathbf{l}}_{ij}) + d^2_{PH}(\mathbf{y}_{ij}, \hat{\mathbf{l}}_{ij}) \right) \right),$$

où $\hat{\mathcal{S}}'$ et $\hat{\mathcal{S}}$ désignent respectivement les droites de la première reconstruction exprimées dans la deuxième base grâce au mouvement estimé, et les droites de la deuxième reconstruction exprimées dans la première base grâce à l'inverse de ce mouvement.

Cette méthode est proche du maximum de vraisemblance. La différence est que les ensembles de droites $\hat{\mathcal{S}}$ et $\hat{\mathcal{S}}'$ sont prédits par transfert à partir de l'autre reconstruction. Cette méthode ne calcule donc pas, conjointement au mouvement, un ensemble de droites parfaitement en correspondance.

8.7 Résultats expérimentaux

Nous validons et comparons nos algorithmes. Nous utilisons des données simulées puis réelles. Nous détaillons ci-dessous les méthodes comparées, l'erreur mesurée et la reconstruction de chaque jeu de caméras et de droites.

Méthodes comparées. Nous comparons les méthodes d'estimation d'une homographie 3D, c'est à dire du mouvement entre deux reconstructions projectives de droites. Les caractéristiques de ces méthodes sont résumées dans le tableau 8.2. Les méthodes comparées sont toutes celles du premier groupe (linéaires et quasi-linéaires), de LIN_3D à 3DLMM_QLIN_2D, et celles du dernier groupe (non-linéaires), NLIN_1, NLIN_2 et le maximum de vraisemblance MLE.

Nous comparons nos estimateurs à des méthodes basées sur des correspondances de points : PTS_LIN_3D et PTS_NLIN_2D. Ces méthodes sont décrites entre autres par Csurka et al. [47]. Elles nécessitent la reconstruction des caméras et de correspondances de points à partir de chaque jeu d'images. Nous effectuons cette reconstruction par ajustement de faisceaux pour chaque jeu d'images, avec une méthode standard. Nous renvoyons au chapitre §3 pour plus de détails. La première méthode est linéaire. Elle estime les paramètres du mouvement en minimisant une erreur algébrique 3D induite par la distance d_{A_VV}, entre des points transférés de la première reconstruction vers la deuxième reconstruction, et les points correspondants de la deuxième

reconstruction. La deuxième méthode est non-linéaire. Elle consiste à minimiser l'erreur de transfert dans les deux jeux d'images, elle est donc équivalente à la méthode NLIN_2 pour les droites. Csurka et al. montrent que c'est, parmi les méthodes qu'ils étudient, la méthode la moins sensible au bruit.

Erreur mesurée. Nous mesurons la qualité d'une estimation en utilisant l'*erreur d'estimation*, équivalente avec nos hypothèses sur la distribution du bruit, à l'erreur de reprojection. La borne inférieure, désignée par BORNE INF., est calculée comme indiqué dans [93, §4].

Nous donnons ci-dessous une méthode pour le calcul de l'erreur de reprojection associée à un mouvement. Le maximum de vraisemblance minimise l'erreur de reprojection, sur les paramètres des caméras et des droites. Cette erreur reflète la qualité de l'estimation. Elle permet de juger de même la qualité d'un mouvement estimé autrement qu'au maximum de vraisemblance. Cependant, les autres algorithmes n'estiment pas les correspondances parfaites de droites, mais seulement les paramètres du mouvement. L'erreur de reprojection n'est donc pas donnée directement. Son calcul nécessite de déterminer des droites se correspondant parfaitement par le mouvement estimé. Ces droites sont estimées individuellement en minimisant l'erreur de reprojection. En pratique, on utilise l'algorithme de Levenberg-Marquardt, avec la représentation orthonormale des droites 3D présentée en §6.4.2, chapitre 6. Une fois les correspondances parfaites obtenues, l'erreur de reprojection est donnée naturellement, en sommant les erreurs de reprojection individuelles. Cet algorithme ne modifie pas le mouvement estimé, mais nous renseigne sur sa qualité, en calculant l'erreur de reprojection associée.

Reconstruction des droites et des caméras. La reconstruction de chaque jeu de caméras et de droites est effectuée par ajustement de faisceaux, c'est à dire par minimisation de l'erreur de reprojection, sur les paramètres des caméras et des droites reconstruites. Nous renvoyons au chapitre 6 pour d'amples détails sur le calcul d'une solution initiale et la procédure d'optimisation non-linéaire.

8.7.1 Données simulées

Scène simulée. Notre scénario expérimental simulé consiste de deux jeux de $n = n\prime = 5$ caméras, observant des segments 3D, choisis au hasard dans une sphère de rayon 1 mètre, par tirage aléatoire de leurs extrémités. Cette simulation est proche de celle faite en §6.5, chapitre 6 pour la reconstruction de droites et de caméras. Pour simuler des images réalistes, nous fixons la distance focale à 1000 (en nombre de pixels). Cette information n'est pas utilisée par la suite. Les extrémités des segments 3D sont projetées dans toutes les images, et leurs positions sont bruitées avec un bruit additif Gaussien centré. Les matrices de projection sont exprimées dans un repère centré sur la première caméra de chaque jeu. Les droites de chaque jeu sont reconstruites indépendamment en minimisant l'erreur de reprojection puis un ajustement de faisceaux est lancé sur chacune des deux reconstructions projectives. Les paramètres par défaut de cette simulation sont $m = 50$ correspondances de droites et une bruit Gaussien d'un écart-type de 1 pixel. Nous varions différents paramètres indépendamment pour déterminer leur influence sur les différentes méthodes.

Résultats. La première expérience, figure 8.6, consiste à varier le niveau de bruit sur la position des extrémités dans les images, de 0 à 2 pixels. Notons qu'une erreur de 2 pixels est élevée par rapport à un positionnement précautionneux de droites dans des images réelles, en particulier lorsque les droites sont déterminées comme le meilleur ajustement à un ensemble de points. Nous observons que l'erreur d'estimation augmente proportionnellement à l'écart-type du bruit sur les extrémités images.

La deuxième expérience, figure 8.7, consiste à changer le nombre de correspondances de droites simulées, de 5 à 105. Notons que, comme établi en §8.2, 5 est le nombre minimum de droites nécessaire pour calculer une homographie 3D. Cependant, les méthodes linéaires et quasi-linéaires 3DLMM_*, basées sur la matrice d'homographie pour droites 3D ne s'appliquent pas à moins de 7 correspondances. Un nombre de 20-100 correspondances de droites peut être obtenu de manière réaliste avec des images réelles, selon la scène considérée. Nous observons que la plupart des méthodes ne réduisent plus l'erreur lorsque le nombre de correspondances dépasse 50, ce qui n'est pas vrai pour les méthodes basées sur une fonction d'erreur 3D. Ces méthodes se stabilisent seulement avec plus de 90 correspondances de droites.

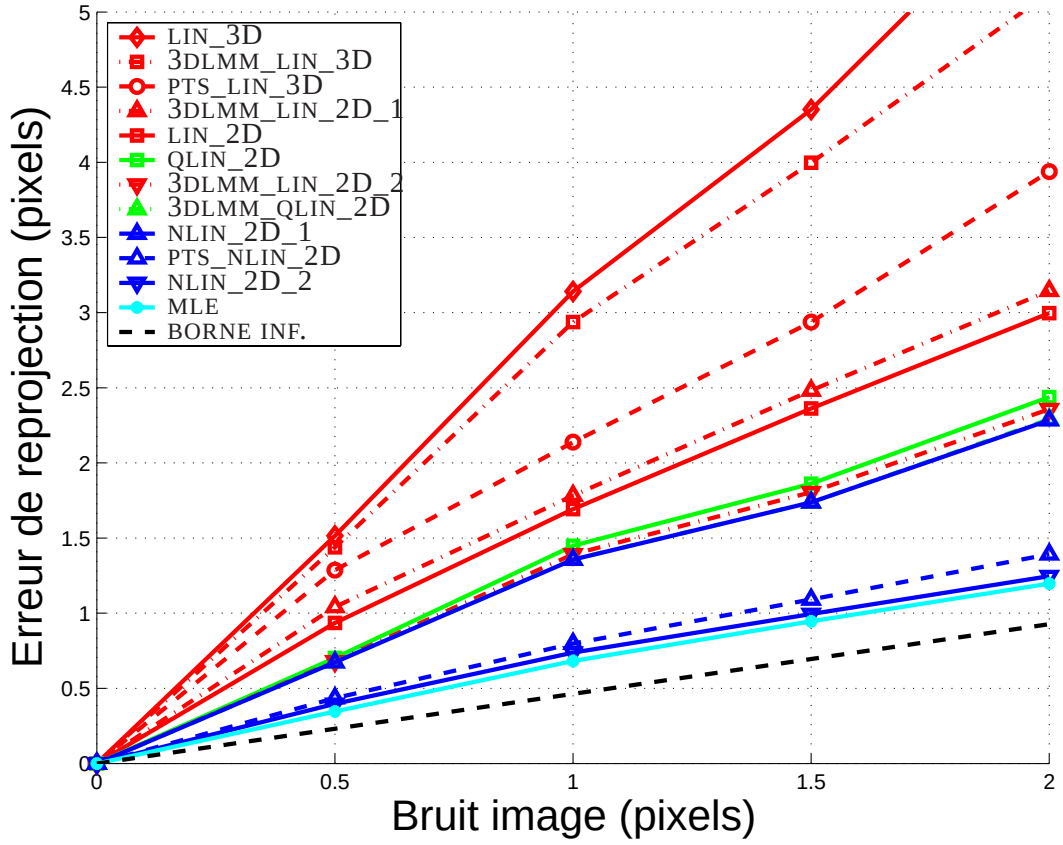


FIG. 8.6 – Erreur de reprojection pour différents niveaux de bruit. Les courbes pour les méthodes 3DLMM_QLIN_2D et NLIN_2D_1 sont confondues.

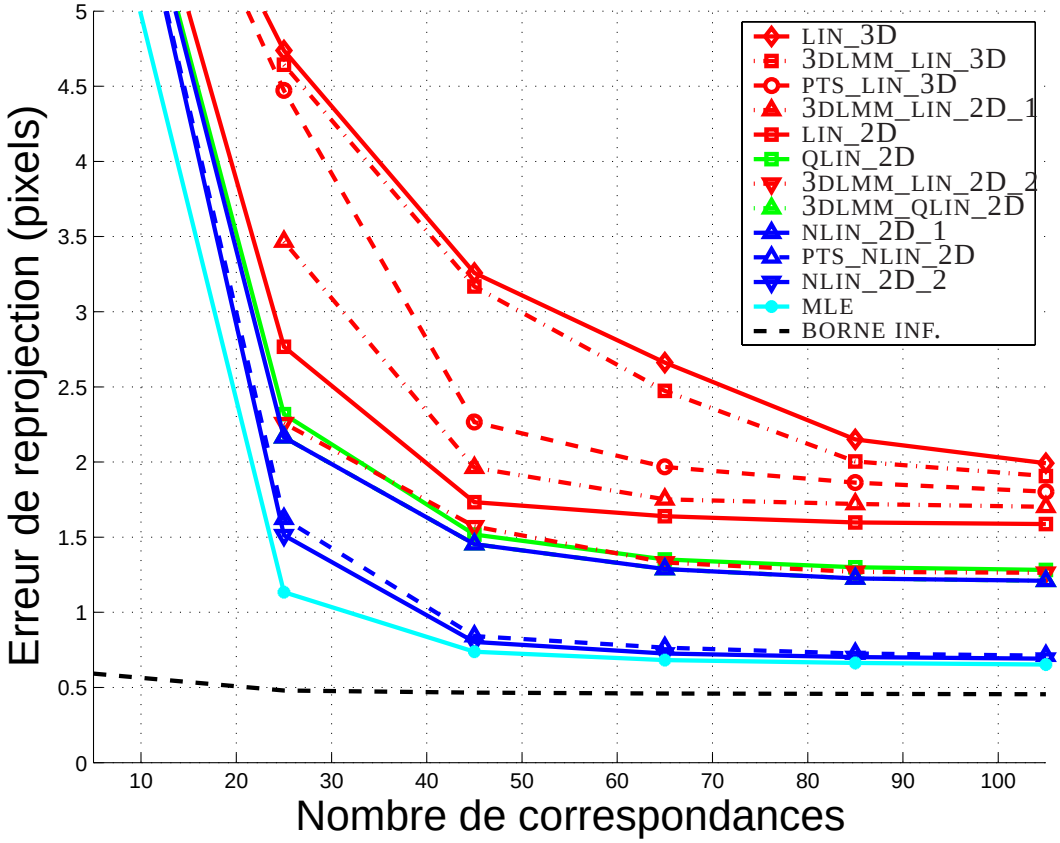


FIG. 8.7 – Erreur de reprojection pour différents nombres de droites. Notons que les méthodes linéaires et quasi-linéaires basées sur la matrice d'homographie pour droites 3D, 3DLMM_*, ne s'appliquent pas à moins de 7 droites. Les courbes pour les méthodes 3DLMM_QLIN_2D et NLIN_2D_1 sont confondues.

Les remarques suivantes concernent les deux expériences. On voit nettement se dégager deux groupes de méthodes sur les graphes des figures 8.6 et 8.7.

Le premier groupe inclut les méthodes linéaires. La meilleure méthode de ce groupe est basée sur un critère 2D, les autres, les méthodes *_3D, basées sur des critères d'erreur exprimés en 3D sont très sensibles au bruit, et ne doivent pas être utilisées. Ce résultat est normal car on travaille en espace projectif. Il n'y a pas de mesure de distance significatif dans un tel espace. La méthode LIN_2D donne des résultats suffisamment fiables pour initialiser les méthodes non-linéaires. Cette méthode est simple à implanter et utilise des données minimales ou redondantes. Elle peut donc aussi être utilisée dans la boucle interne d'un estimateur robuste par échantillonnage aléatoire, tel RANSAC [64].

L'autre groupe de méthodes est constitué des méthodes quasi-linéaires et non-linéaires, minimisant des critères d'erreur basés sur des distances Euclidiennes et ayant donc un sens physique. Les méthodes non-symétriques, QLIN_2D, 3DLMM_QLIN_2D et NLIN_1 donnent des résultats très similaires. Notons que les méthodes QLIN_2D et 3DLMM_QLIN_2D sont basées sur une repondération itérative du système linéaire utilisé par les méthodes LIN_2D et 3DLMM_LIN_2D respectivement. Elles s'implantent donc très simplement. Les méthodes basées sur des fonctions de coût symétriques par rapport aux rôles joués par les deux jeux d'images, NLIN_2, PTS_NLIN_2D et MLE, donnent des résultats meilleurs, et légèrement différents. Le maximum de vraisemblance, méthode MLE, minimise l'erreur d'estimation car il implique les paramètres des droites reconstruites, et il est donc normal qu'il donne un résultat meilleur. Il est suivi de près par NLIN_2 et PTS_NLIN_2D, basées respectivement sur des droites et des points. Lorsque le temps de calcul est important, la méthode NLIN_2 doit être préférée. On observe sur la figure 8.7 que le maximum de vraisemblance ne tend pas asymptotiquement vers la borne inférieure. Ce phénomène est dû au fait que les caméras reconstruites préalablement à l'alignement des reconstructions, ne sont pas changées par le maximum de vraisemblance. Ces paramètres de la scène sont donc figés et ne permettent pas de réduire l'erreur jusqu'à la borne inférieure.

Ces graphes mettent en évidence la différence entre les méthodes basées sur des mesures d'erreur entre des points reprojetés et des droites observées, par exemple les méthodes NLIN_2D et QLIN_2D, et celles utilisant une erreur exprimée entre des droites reprojetées et des points mesurés, par exemple la méthode 3DLMM_QLIN_2D. On observe que cette dernière donne de meilleurs résultats. Ceci est dû au fait que la forme de la fonction de coût est plus proche de celle du maximum de vraisemblance que pour les deux autres méthodes.

8.7.2 Données réelles

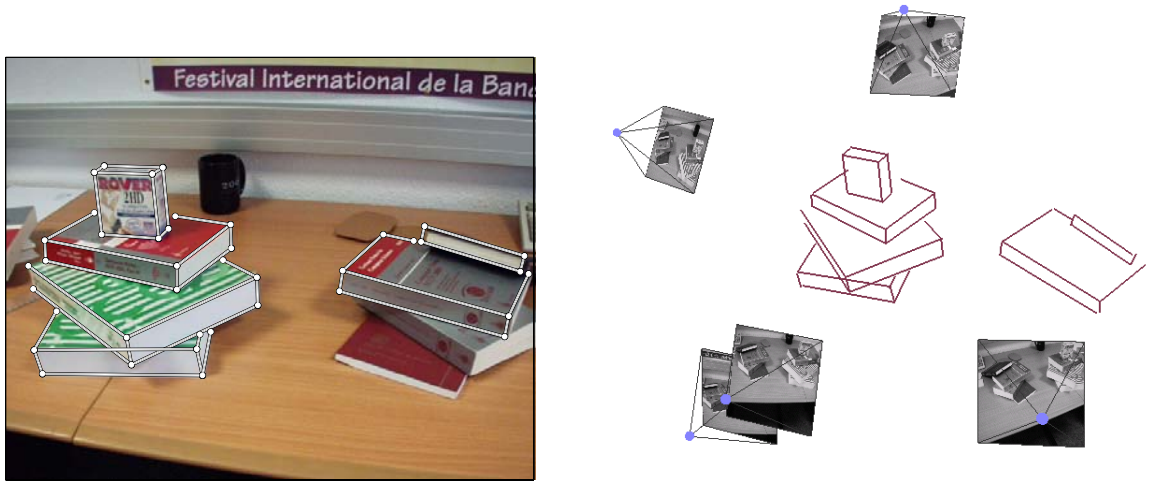
Nous testons nos algorithmes sur trois cas issus d'images réelles. Deux scénarios sont considérés. Le premier, les séquences des livres et les séquences des boîtes, consiste à aligner deux reconstructions métriques, via le calcul d'une similitude. Nous comparons les algorithmes basés sur la matrice de mouvement usuelle, et ceux basés sur la matrice de mouvement pour droites 3D, aux méthodes non-linéaires sur les séquences des livres et des boîtes respectivement. Le recouvrement entre deux séquences, c'est à dire le nombre de correspondances de droites entre les deux reconstructions, n'est pas élevé. Dans de telles configurations géométriques, le nombre de correspondances de droites est important. Pour ces deux expériences, nous utilisons respectivement 21 puis 15 correspondances. L'alignement peut donc être instable.

Le deuxième scénario considère deux paires d'images non calibrées, les séquences de la pièce de bateau, prises avec une tête stéréoscopique rigide. L'alignement est effectué en estimant une homographie 3D. Le recouvrement entre les deux séquences est grand, et même si seulement 21 correspondances de droites sont utilisées, on s'attend à une estimation précise du mouvement. Nous appliquons une technique de calibrage en ligne stéréoscopique permettant de transformer les reconstructions dans un espace métrique. Notons que dans ces expérimentations, la localisation et la mise en correspondance des droites sont effectuées manuellement.

8.7.2.1 Séquences des livres

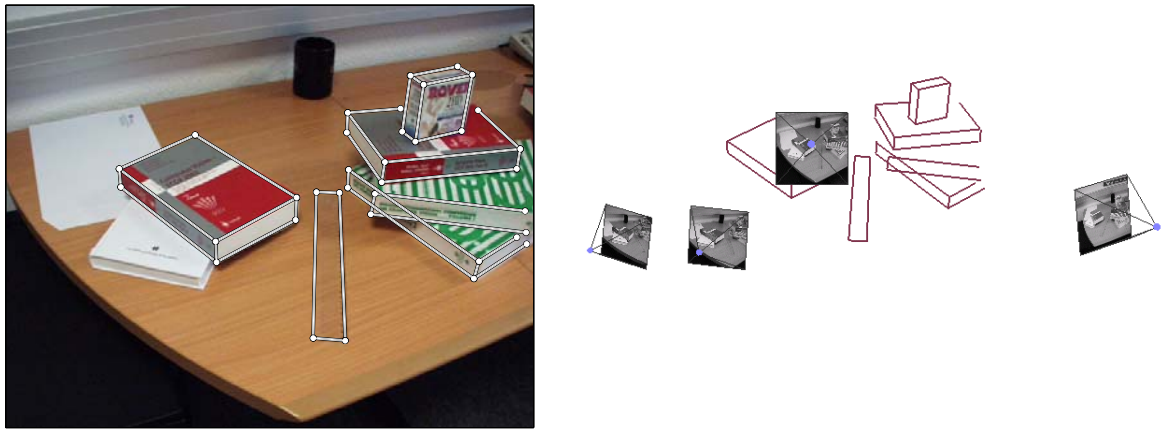
Cette expérimentation consiste à aligner les deux reconstructions obtenues à partir des séquences des livres I et II. Ces deux reconstructions métriques de droites et de caméras sont obtenues à partir de la technique d'ajustement de faisceaux proposée au chapitre 6. Nous renvoyons aux sections 6.5.2.1 et 6.5.2.2 pour plus de détails

sur le déroulement de ces reconstructions. Les figures 8.8 et 8.9 montrent une image de la séquence originale et une vue du modèle 3D reconstruit pour les séquences de livres I et II respectivement. On observe dans chaque



(a) (b)

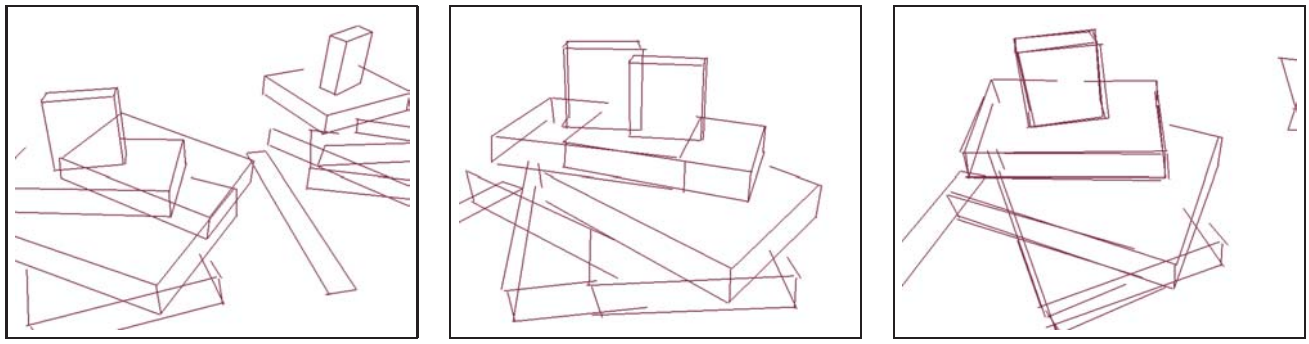
FIG. 8.8 – Une des 5 images de la séquence des livres I (a) et une vue du modèle 3D reconstruit, consistant des 5 caméras et de 45 droites. L'erreur de reprojection après ajustement de faisceaux est de 0,94 pixels.



(a) (b)

FIG. 8.9 – Une des 4 images de la séquence des livres II (a) et une vue du modèle 3D reconstruit, consistant de 4 caméras et de 40 droites. L'erreur de reprojection après ajustement de faisceaux est de 0,51 pixels.

séquence une pile de livres, plus d'autres éléments (règle, livres, ...). La pile de livres est commune aux deux séquences, et fournit 21 correspondances de droites entre les deux reconstructions. Notons que ces droites sont localisées de manière proche dans l'espace, ce qui rend la tâche d'alignement instable. La figure 8.10 (a) montre la visualisation des deux reconstructions sans alignement, c'est à dire leur simple superposition, afin de donner visuellement une idée du mouvement à estimer. La figure 8.10 (b) montre le résultat de l'alignement après l'estimation d'une similitude avec l'algorithme 3DLMM_LIN_3D, basé sur un critère 3D. Finalement, la figure 8.10 (c) montre le résultat de l'alignement après l'extraction d'une similitude, par l'algorithme donné en §8.5.3.1, à partir d'une affinité estimée avec l'algorithme AFF_QLIN_2D, basé sur une erreur entre des points prédits et des droites observées, exprimée dans le deuxième jeu d'images. L'algorithme d'extraction consiste principalement à calculer la matrice de rotation à l'échelle la plus proche, au sens de la norme de Frobenius, de la partie (3×3) supérieure gauche de la matrice d'affinité. L'erreur de reprojection, dans le second jeu d'images, est 15,3 pixels



(a) (b) (c)

FIG. 8.10 – (a) montre la visualisation des deux reconstructions sans alignement préalable, (b) montre le résultat de l'alignement après le calcul d'une similitude avec l'algorithme 3DLMM_SIM_LIN_3D et (c) après l'extraction d'une similitude, par l'algorithme donné en §8.5.3.1, à partir d'une affinité estimée avec l'algorithme AFF_QLIN_2D.

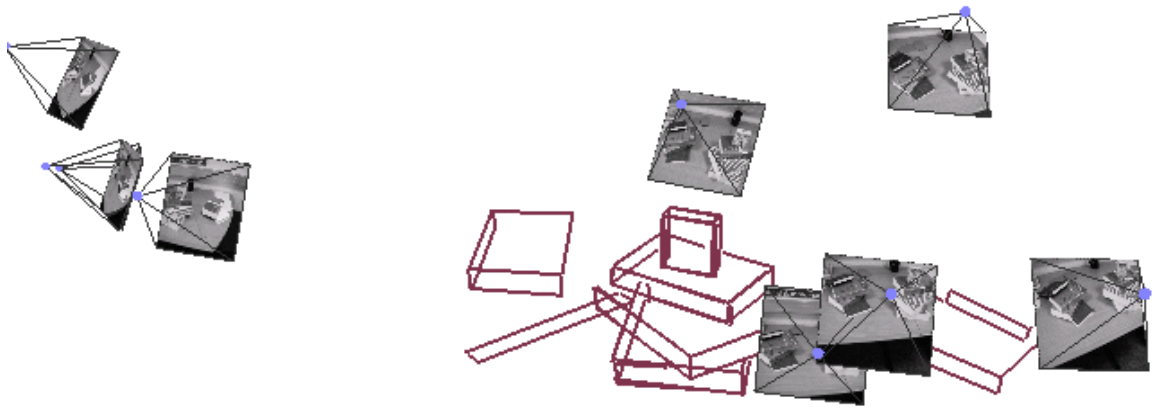


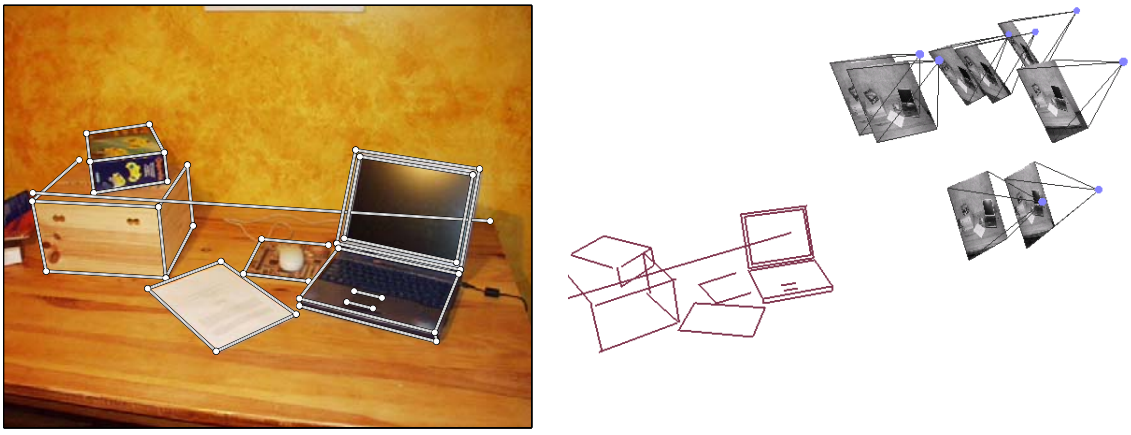
FIG. 8.11 – Alignement des deux séquences des livres. La méthode basée sur le calcul d'une affinité avec l'algorithme AFF_QLIN_2D puis extraction d'une similitude, et le maximum de vraisemblance MLE, produisent des résultats visuellement très proches.

pour la méthode 3DLMM_LIN_3D et 3,7 pixels pour la méthode AFF_QLIN_2D suivie de l'extraction d'une similitude. L'erreur de reprojection sur les deux jeux d'images est 14,9 pixels pour la méthode 3DLMM_LIN_3D et 4,8 pixels pour la méthode AFF_QLIN_2D suivie de l'extraction d'une similitude. L'alignement produit par la méthode 3DLMM_LIN_3D est clairement moins bon que celui produit par AFF_QLIN_2D suivie de l'extraction d'une similitude, et il est même très mauvais. La raison principale est que cette méthode est basée sur la comparaison de droites 3D. Or, il n'existe pas de mesure de distance Euclidienne entre droites 3D, et il est peu vraisemblable que le critère utilisé par cette méthode ait un sens physique. La méthode AFF_QLIN_2D, en revanche, minimise un critère ayant un sens physique, même s'il est différent de l'erreur de reprojection. En effet, cette méthode minimise la distance entre des points sur les droites de la première reconstruction, transférés puis reprojetés dans le second jeu d'images, avec les droites observées dans ces images. Ceci est différent de l'erreur de reprojection, exprimée entre des droites prédites et des points mesurés.

La figure 8.11 montre les 9 caméras et les 64 droites obtenues après alignement. Le maximum de vraisemblance donne une erreur de reprojection de 4,3 pixels. Il produit un résultat visuellement très proche de la méthode basée sur le calcul d'une affinité avec l'algorithme AFF_QLIN_2D puis extraction d'une similitude. La différence principale est qu'il permet d'obtenir des correspondances parfaites de droites, et donc une droite unique lors de l'alignement des correspondances.

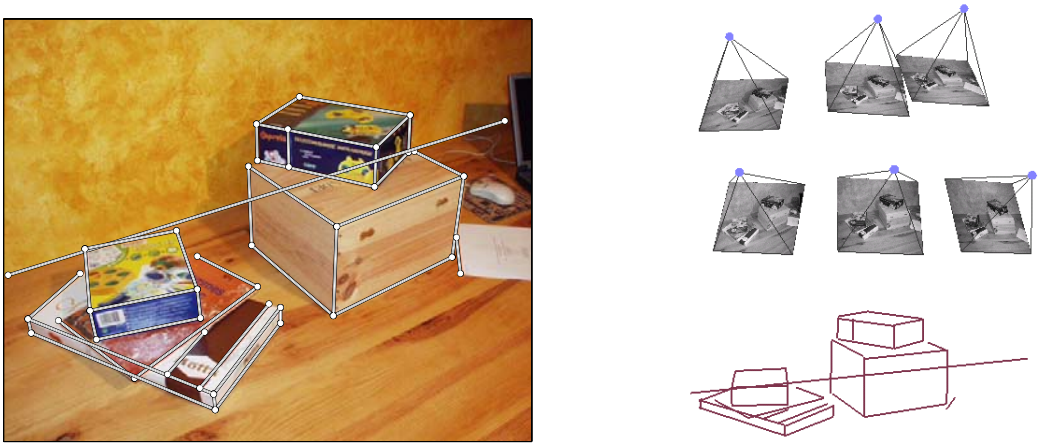
8.7.2.2 Séquences des boîtes

Cette expérimentation est basée sur un scénario semblable à celui de l'expérimentation précédente sur les séquences des livres : le but est d'aligner deux reconstructions métriques d'une même scène. Les caméras sont calibrées et leurs positions sont reconstruites à partir de correspondances de points. Les figures 8.12 et 8.13 montrent une image de la séquence originale et une vue du modèle 3D reconstruit, pour les séquences des boîtes I et II respectivement. Nous renvoyons au §§6.5.2.3 et 6.5.2.4 pour des détails sur l'estimation de ces modèles 3D.



(a) (b)

FIG. 8.12 – Une des 8 images de la séquence des boîtes I (a) et une vue du modèle 3D reconstruit, consistant de 8 caméras et 40 droites. L'erreur de reprojection après ajustement de faisceaux est de 0,82 pixels.



(a) (b)

FIG. 8.13 – Une des 6 images de la séquence des boîtes II (a) et une vue du modèle 3D reconstruit, consistant de 6 caméras et 40 droites. L'erreur de reprojection après ajustement de faisceaux est de 0,87 pixels.

La scène observée est constituée de deux piles de boîtes et d'un ordinateur portable. Dans la séquence des boîtes I, la pile de boîtes de gauche n'est pas visible, alors que dans la séquence des boîtes II, c'est l'ordinateur portable qui n'est pas visible. En conséquence, même si chaque reconstruction est constituée d'un jeu de 40 droites, nous n'avons que 15 correspondances de droites, relatives à la pile de boîtes du milieu et donc très proches dans l'espace. L'alignement est effectué comme suit. Concernant les méthodes linéaires, on utilise

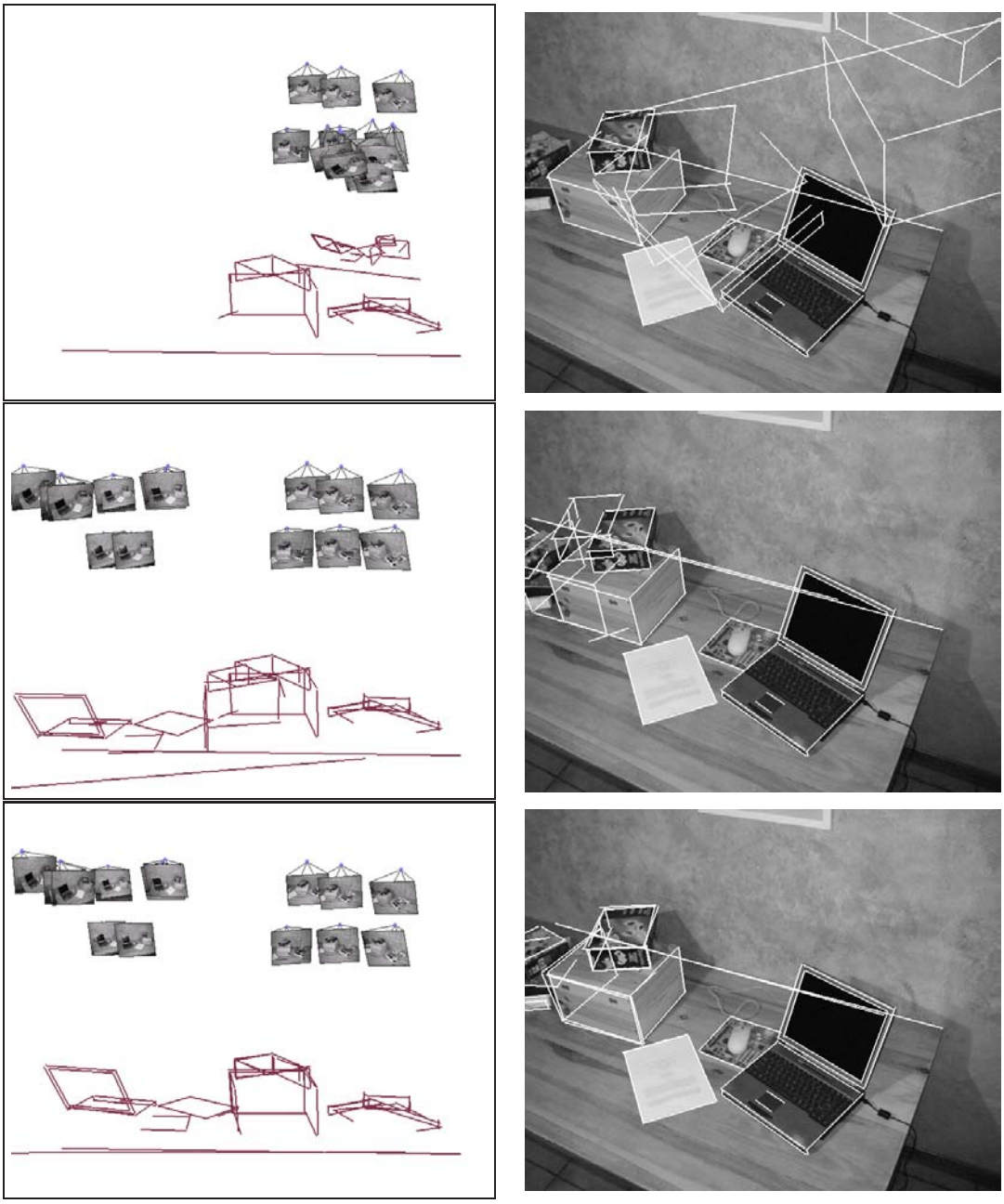


FIG. 8.14 – Les deux reconstructions de droites et de caméras, de haut en bas : sans alignement, alignement avec les méthodes linéaires puis alignement avec les méthodes quasi-linéaires et non-linéaires. Les méthodes utilisées sont basées sur l'estimation d'une homographie avec les méthodes basées sur la matrice d'homographie pour droites 3D, 3DLMM_*, puis extraction d'une similitude avec l'algorithme donné en §7.4.3, chapitre 7. La colonne de gauche montre les reconstructions alignées, et la colonne de droite, la reprojection de l'alignement dans une image originale.

les méthodes d'estimation d'une homographie basées sur la matrice d'homographie pour droites 3D, notées 3DLMM_*, voir §8.5.1.2. Ces méthodes sont basées sur le calcul d'une matrice (6×6) de forme générale, dont on extrait une homographie. Cette dernière étape d'extraction de l'homographie est ignorée, et remplacée par l'extraction d'une similitude, par l'algorithme décrit en §7.4.3, chapitre 7. Concernant les méthodes non-linéaires, nous avons précisé, en §8.6, qu'elles ne dépendaient pas intrinsèquement du type de mouvement estimé : seule la paramétrisation de ce dernier change, mais la fonction de coût reste la même.

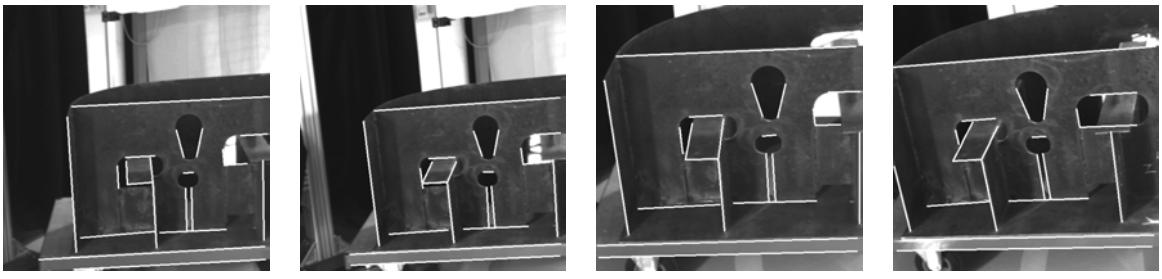
La figure 8.14 montre les résultats obtenus. Ces derniers peuvent être classés dans deux catégories. Les méthodes linéaires donnent de très mauvais résultats, traduits par des erreurs de reprojection élevées (voir ci-dessous) et un alignement visuel décalé, alors que les méthodes quasi-linéaires et non-linéaires donnent de bons résultats. Nous avons mesuré l'erreur de reprojection (dans les deux jeux d'images) pour les différentes méthodes. Le tableau suivant donne ces mesures, où chaque méthode est indiquée par le nom de la méthode utilisée pour estimer la matrice (6×6) dont est ensuite extraite la similitude :

méthode	erreur de reprojection (pixel)
3DLMM_LIN_3D	14,49
3DLMM_LIN_2D_1	15,10
3DLMM_LIN_2D_2	13,28
NLIN_2D_1	2,95
QLIN_2D	2,91
NLIN_2D_2	1,76
MLE	1,24

Ces mesures confirment les observations précédentes : l'erreur de reprojection est de l'ordre de 10 pixels pour les méthodes linéaires, ce qui est beaucoup, alors qu'elle est de l'ordre de quelques pixels pour les méthodes quasi-linéaires et non-linéaires. Ces résultats s'expliquent par le fait que les correspondances de droites ne sont pas bien réparties dans l'espace, ce qui induit un système instable. Comme les méthodes linéaires nécessitent plus d'information que les autres méthodes pour produire un bon résultat, le fait que seules 15 correspondances soient disponibles entraîne une mauvaise estimation du mouvement.

8.7.2.3 Séquences de la pièce de bateau

Cette expérimentation repose sur un scénario différent des deux précédents, dans le sens où le recouvrement entre les deux reconstructions est maximal : toutes les droites reconstruites sont en correspondance. Les images ont été prises avec une tête stéréoscopique rigide faiblement calibrée. Ces images sont montrées en figure 8.15. Faiblement calibrée signifie que la géométrie épipolaire entre les deux caméras est connue. Le fait que la tête stéréoscopique soit rigide implique que cette géométrique épipolaire est identique pour les deux paires d'images.



paire 1 paire 2

FIG. 8.15 – Les deux paires d'images constituant les séquences de la pièce de bateau. Un total de 21 droites, montrées en blanc, sont visibles dans toutes les images.

Reconstruction projective des caméras et des droites. Nous estimons la géométrie épipolaire à partir de correspondances de points, en utilisant un estimateur au maximum de vraisemblance, voir le chapitre 4. A partir de cette matrice fondamentale, nous définissons une base de reconstruction projective canonique pour chaque paire d'images. Nous obtenons ensuite les coordonnées de Plücker des droites issues de chaque paire d'images en calculant l'intersection des deux plans de vue, voir §2.12, chapitre 2.

Alignement projectif. Nous utilisons les méthodes basées sur la matrice d'homographie pour droites 3D, voir §8.5.1.2, pour estimer l'homographie permettant d'aligner les deux reconstructions. La figure 8.16 montre le résultat des droites transférées de la première reconstruction vers la deuxième, puis reprojetées sur la deuxième paire d'images. Ces visualisations confirment que les méthodes quasi-linéaires et non-linéaires donnent de

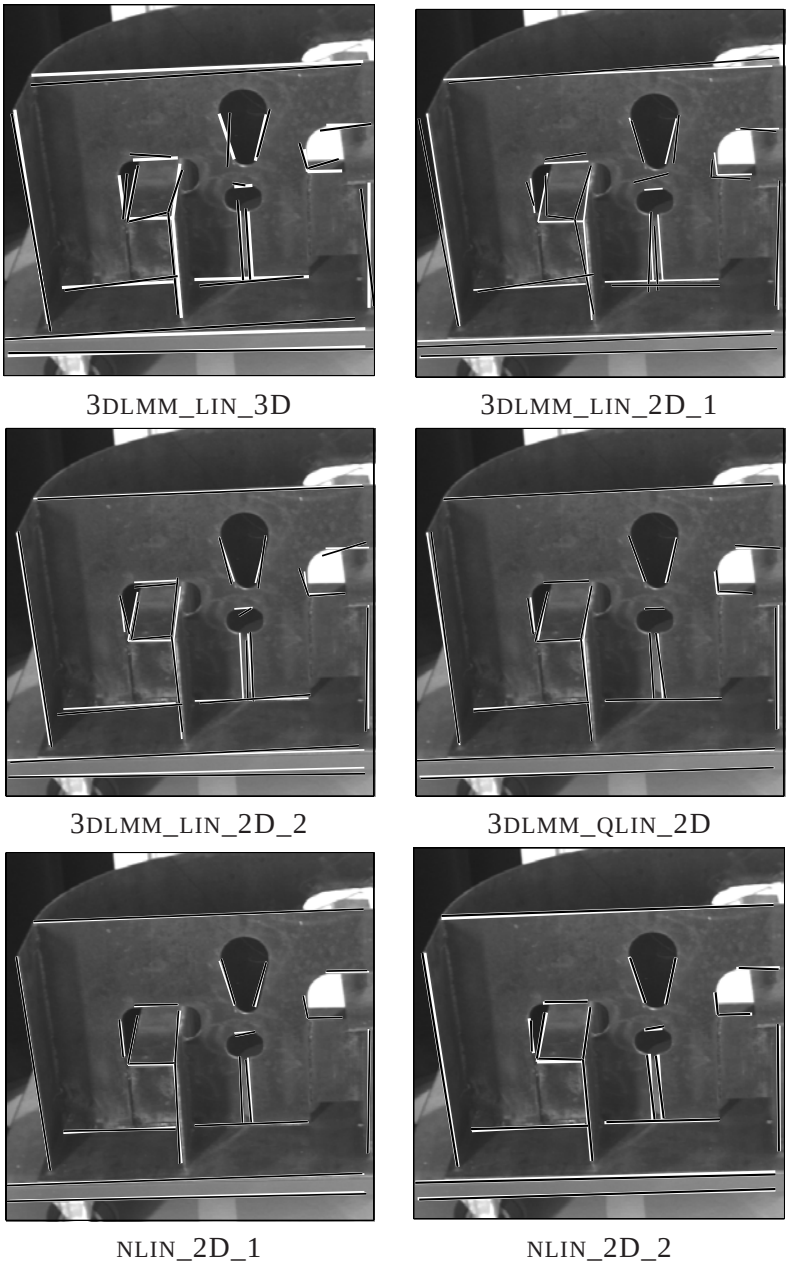


FIG. 8.16 – Reprojection sur la deuxième paire d'images des droites de la première reconstruction projective, transférées dans le repère de la deuxième par l'homographie estimée. Les droites reprojetées sont en noir alors que les droites originales sont en blanc.

meilleurs résultats que les méthodes linéaires. Cependant, la différence entre ces méthodes est réduite par rapport aux expérimentations précédentes, par le fait que plus de correspondances de droites sont disponibles. On peut noter que les résultats apparaissent visuellement moins bon pour NLIN_2D_2 que pour NLIN_2D_1 car le premier minimise une erreur de transfert dans les deux jeux d'images, alors que le deuxième ne la minimise que dans le deuxième jeu d'image, celui que nous visualisons. Nous avons mesuré l'erreur de reprojection dans les deux jeux d'images :

méthode	erreur de reprojection (pixel)
3DLMM_LIN_3D	3,45
3DLMM_LIN_2D_1	3,36
3DLMM_LIN_2D_2	2,83
NLIN_2D_1	2,05
QLIN_2D	1,93
NLIN_2D_2	1,53
MLE	1,21

Ces résultats confirment les observations précédentes : même s'il y a quelques différences entre les méthodes, elle donnent toutes un résultat correct, dans le sens où l'erreur de reprojection est d'un ordre raisonnable (quelques pixels).

Calibrage en ligne. Le fait que la tête stéréoscopique utilisée pour prendre les paires d'images soit rigide nous permet d'utiliser une technique de calibrage en ligne stéréoscopique. Nous utilisons l'homographie estimée avec la méthode NLIN_2D_1 pour calibrer les caméras, avec la méthode décrite dans [100], en supposant que les caméras n'ont pas de distorsion entre les axes et un rapport d'aspect unitaire. Cet algorithme calcule une matrice de passage projectif-métrique notée H_u, c'est à dire une matrice (4×4) permettant de convertir la reconstruction projective en une reconstruction métrique. La structure de cette matrice est explicitement donnée en §7.5, chapitre 7, où nous donnons la matrice de passage projectif-métrique pour droites 3D correspondante. Cette matrice, de taille (6×6), nous permet de convertir directement les coordonnées de Plücker représentant les droites reconstruites, du projectif vers le métrique. La figure 8.17 montre la reconstruction métrique ainsi obtenue.

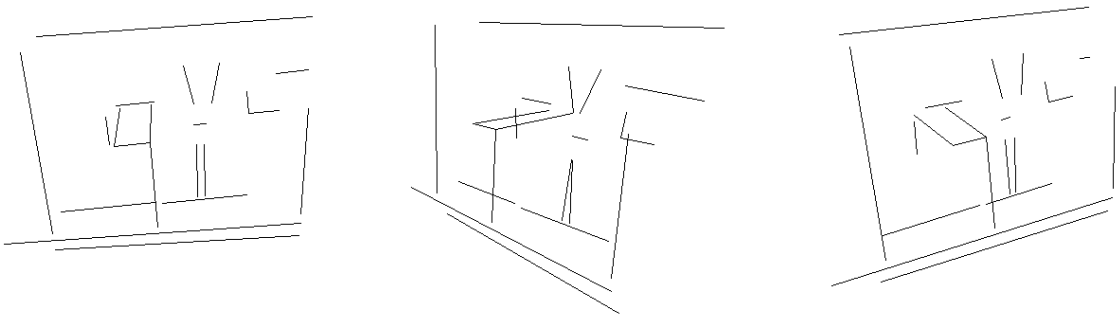


FIG. 8.17 – Les droites des séquences de la pièce de bateau, reconstruites en espace métrique, après calibrage en ligne des caméras à l'aide d'une technique de calibrage en ligne stéréoscopique.

Nous comparons les paramètres intrinsèques retrouvés par calibrage en ligne pour la caméra de gauche, à ceux estimés par calibrage statique à l'aide d'une mire [63] :

paramètre	calibrage en ligne	calibrage hors ligne	% erreur
f	1514,22	1461,02	3,51
u_0	252,93	267,98	5,95
v_0	250,68	241,03	3,84

où (u_0, v_0) sont les coordonnées du point principal dans l'image, et f est la distance focale, en pixels. La précision obtenue est raisonnable, et équivalente à ce que l'on obtient souvent avec des points à l'issue d'un calibrage en ligne.

8.8 Conclusions et perspectives

Conclusions. Nous avons abordé le problème de l'estimation d'un mouvement 3D, à partir de correspondances de droites, provenant de deux reconstructions indépendantes d'une scène rigide. Nous avons examiné les cas de reconstructions projectives, affines, métriques et Euclidiennes. Après avoir établi les cas minimaux, nous formulons le maximum de vraisemblance, minimisant l'erreur de reprojection. Les méthodes "auto-initialisantes" proposées, linéaires ou quasi-linéaires, sont de deux types. Le premier type implique la détermination directe d'une matrice de mouvement usuelle (4×4). L'idée est de représenter différemment les droites des deux reconstructions. Celles de la première reconstruction sont représentées par des points, et celles de la deuxième par des plans. Cette représentation permet d'obtenir des équations linéaires sur les coefficients de la matrice de mouvement usuelle. L'erreur minimisée est exprimée en 3D, ce qui n'a pas de sens car il n'existe pas de distance Euclidienne entre deux droites. Ces méthodes sont ensuite modifiées, afin de minimiser une fonction de coût exprimée dans les images. Cette fonction de coût est "duale" à l'erreur de reprojection, dans le sens où la différence entre des points prédits et des droites observées est minimisée. Le deuxième type de méthodes est basé sur une matrice de mouvement pour droites 3D. Les droites sont alors représentées par leurs coordonnées de Plücker. Une erreur de transfert proche de l'erreur de reprojection est minimisée, mais l'algorithme nécessite une étape finale d'extraction des paramètres du mouvement, introduisant un biais dans l'estimation. Notons que l'erreur de reprojection et sa version "duale" sont aussi observées au chapitre 6 dans le cas de la reconstruction de droites : comme ici, les méthodes utilisant une représentation basée sur des points minimisent la version "duale", et nos méthodes basées sur les coordonnées de Plücker minimisent une erreur de transfert similaire à l'erreur de reprojection, mais nécessitent une étape de correction, liée à la contrainte de Plücker. Finalement, nous avons proposé des estimateurs non-linéaires, minimisant différentes erreurs de transfert, avec une paramétrisation adaptée du mouvement.

Nous avons validé et comparé nos méthodes sur des données simulées et réelles. Nos conclusions principales par rapport à ces expérimentations sont les suivantes. Les méthodes basées sur des contraintes 3D donnent de mauvais résultats, et ne doivent pas être utilisées, sauf dans les cas minimaux où une solution exacte est déterminée, par exemple dans la boucle interne d'un estimateur robuste par échantillonnage aléatoire. Les méthodes basées sur une matrice de mouvement pour droites 3D donnent des résultats légèrement meilleurs que celles basées sur une matrice de mouvement usuelle, mais sont plus compliquées à implanter.

Pour résumer, nous recommandons l'utilisation d'une méthode non-linéaire pour l'estimation du mouvement à partir de correspondances de droites : la méthode NLIN_2D_2 (minimisation d'une erreur de transfert Euclidienne symétrique) si le temps de calcul est important, et le maximum de vraisemblance MLE (minimisation de l'erreur de reprojection) dans le cas contraire. Ces méthodes nécessitent une initialisation. Nous recommandons la méthode 3DLMM_QLIN_2D (minimisation d'une erreur de transfert Euclidienne non symétrique), basée sur une matrice de mouvement pour droites 3D.

Perspectives. Nous proposons quelques perspectives de travail pour terminer ce chapitre. La première est l'analyse des configurations singulières, c'est à dire des configurations de correspondances de droites ne définissant pas un mouvement 3D unique. Notons que les études des configurations singulières pour la reconstruction de droites et de caméras menées par Buchanan [38] et Navab et Faugeras [152] peuvent fournir un point de départ.

Deux autres perspectives de travail concernent les méthodes proposées. Les méthodes basées sur une matrice de mouvement usuelle dépendent du choix des points utilisés pour représenter les droites. En pratique, nous avons remarqué que le choix de ces points influençait peu le résultat. Cependant, on note que ces méthodes peuvent être rendues indépendantes au choix des points représentant les droites, en pondérant les différentes

erreurs dans les images, par la covariance de la droite observée[7].

Les estimateurs basés sur une matrice de mouvement pour droites 3D minimisent une erreur de transfert proche de l'erreur de reprojection, mais sont biaisés par l'étape finale d'extraction du mouvement. Comme nous l'avons fait dans le cas de la reconstruction de droites, pour la contrainte de Plücker, nous pensons qu'il est possible de linéariser les contraintes de consistance définissant une matrice de mouvement pour droites 3D, et de les intégrer à un estimateur quasi-linéaire, ce qui permettrait de réduire le biais introduit par l'étape d'extraction, et probablement d'accélerer la convergence.

Nous pensons que la principale application pratique de ces méthodes est la reconstruction à partir de longues séquences d'images par les méthodes hiérarchiques de Fitzgibbon et Zisserman [66] et Nistér [155]. Ces méthodes sont basées sur la reconstruction à partir de sous-séquences, puis la fusion des reconstructions. Chaque étape de fusion nécessite d'aligner deux reconstructions, ce qui est effectué à l'aide de correspondances de points. Il serait intéressant de tester, sur des séquences réelles, le gain en précision apporté par l'utilisation conjointe de droites.

Finalement, une perspective de travail est l'étude des configurations de points sur une droite. Bouguet et Perona [37] définissent et étudient une primitive point-sur-droite, qu'ils appelent "P-Line" en Anglais. Une possibilité serait de considérer plusieurs points sur une droite, une situation couramment observée en environnement humain.

[7] Merci à Wolfgang Förstner de l'institut de photogrammétrie à l'université de Bonn pour cette suggestion.

CHAPITRE

9

CONCLUSIONS, PERSPECTIVES ET PROBLÈMES OUVERTS

Cette thèse expose nos contributions aux domaines de la vision 3D par ordinateur et de la photogrammétrie, et plus particulièrement au problème de la reconstruction de modèles 3D à partir d'images. Ce chapitre a pour but de donner nos conclusions par rapport aux méthodes et concepts proposés, des perspectives de travail concernant la reconstruction de modèles 3D, ainsi qu'une discussion sur des problèmes ouverts.

9.1 Conclusions

Le problème de la reconstruction de modèles 3D à partir d'images repose sur différentes tâches de reconstruction. Nous donnons ci-dessous nos conclusions par rapport aux contributions apportées dans cette thèse sur certaines de ces tâches, puis sur des concepts plus globaux émergeant de nos travaux.

- ▷ **Géométrie de deux caméras non calibrées et estimation non-linéaire de la matrice fondamentale.** Nous avons étudié la géométrie algébrique et la paramétrisation de deux caméras. La méthode est spécifique au cas de caméras non calibrées, car le problème est différent (et était résolu) pour le cas de caméras calibrées. Notre paramétrisation, la *représentation orthonormale de la matrice fondamentale*, permet d'effectuer l'ajustement de faisceaux de manière *minimale* avec des expressions mathématiques simples. Nous montrons qu'elle permet d'inclure de l'information connue a priori (par exemple la position d'un épipôle). Nous en discutons les avantages et inconvénients. Nous pensons qu'elle fournit un moyen plus simple et élégant que les méthodes existantes pour l'optimisation non-linéaire minimale de deux caméras non calibrées ;
- ▷ **Reconstruction de plans, de points et de caméras.** Nous avons examiné l'inférence d'un modèle 3D dont la structure est constituée par des points soumis à des contraintes de coplanarité. Les aspects initialisation et paramétrisation pour l'ajustement de faisceaux sont examinés. Ces contributions sont valables pour des caméras calibrées et non calibrées. Notre méthode est optimale, et une simple hypothèse sur les contraintes de coplanarité permet d'obtenir une solution algorithmique efficace pour l'ajustement de faisceaux. Nous pensons que notre approche fournit un moyen efficace pour la reconstruction de scènes planes par morceaux. Elle est plus simple que la plupart des approches optimales existantes. Son défaut principal est d'être spécifique aux contraintes de coplanarité : son extension à d'autres types de contraintes est difficile. Nous avons par ailleurs étudié, comme une contribution annexe de la thèse, la détection de plans, dont le but est d'automatiser les méthodes de reconstruction de plans ;
- ▷ **Reconstruction de droites et de caméras.** Nous avons étudié l'inférence d'un modèle 3D dans le cas où la structure est constituée de droites. Nous examinons les problèmes d'initialisation et de paramétrisation. Nos propositions sont valables dans le cas de caméras calibrées et non calibrées. Nous proposons plusieurs algorithmes de reconstruction initiale, basés sur des fonctions de coût dont la forme est proche de l'erreur de reprojection, contrairement aux méthodes existantes. Nous étudions ensuite la représentation des droites 3D, et proposons la *représentation orthonormale des droites 3D*, une paramétrisation permettant de conduire l'ajustement de faisceaux de manière minimale. Nous montrons qu'elle permet d'inclure de l'information connue a priori de manière simple (par exemple, la direction de la droite). Nous discutons les avantages et inconvénients de notre approche. Nous pensons qu'elle fournit un moyen plus simple et élégant que les méthodes existantes pour l'optimisation non-linéaire minimale d'une droite 3D ;
- ▷ **Les matrices de mouvement pour droites 3D.** Nous menons une étude théorique de l'application d'un mouvement 3D à des droites 3D exprimées en coordonnées de Plücker. Nous mettons en évidence les *matrices de mouvement pour droites 3D*, et étudions les liens avec la représentation usuelle du mouvement. Nous utilisons une approche stratifiée, et formulons les propriétés de ces matrices pour plusieurs niveaux de calibrage des caméras – en espaces projectif, affine, métrique et Euclidien ;
- ▷ **Alignement de reconstructions de droites.** Nous étudions l'alignement de modèles 3D à l'aide de correspondances de droites, contrairement au cas traditionnel, où seules des correspondances de points sont utilisées. Notre approche est stratifiée. Nous abordons les problèmes de l'initialisation et de l'estimation optimale. Nous proposons de nombreuses méthodes d'initialisation, basées sur différentes représentations du mouvement, notamment les matrices de mouvement pour droites 3D.

Nos contributions ont été guidées par les concepts suivants : l'utilisation de fonctions de coût appropriées pour les méthodes d'initialisation et la conception de paramétrisations adaptées pour l'ajustement de faisceaux. Nous avons défini un ensemble de critères permettant de concrétiser ces notions. Nous avons remarqué une forte adéquation entre ces critères et les performances des méthodes, mesurées expérimentalement. Le concept de la représentation orthonormale revient à deux reprise : pour la paramétrisation de la matrice fondamentale et celle des droites 3D à partir des coordonnées de Plücker. Ces deux entités algébriques ont en commun d'être

définies à un facteur multiplicatif près et soumises à une contrainte de consistance interne, ce qui rend difficile leur paramétrisation pour l'optimisation non-linéaire. La représentation orthonormale permet d'effectuer une mise à jour locale basée sur le nombre minimum de paramètres. Il est possible d'imaginer des représentations orthonormales pour d'autres entités algébriques, pour lesquelles il n'existe pas de paramétrisation minimale simple[1]. Le concept plus général est de trouver une sur-paramétrisation de l'entité, permettant une mise à jour minimale, garantissant les contraintes internes. Nous pensons que la représentation orthonormale constitue un outil puissant de paramétrisation pour l'optimisation non-linéaire.

9.2 Perspectives

Les différentes tâches de reconstruction permettant la création de modèles 3D à partir d'images possèdent des solutions plus ou moins abouties. Il existe des logiciels de reconstruction automatique, tels `boujou` de `2d3` ou `MatchMover` de `Realviz`, et semi-automatique, tels `canoma` ou `photomodeller`. A l'issue de cette thèse, nous pensons cependant que les solutions à certaines tâches de reconstruction peuvent être améliorées. Les commentaires suivants concernent des tâches de reconstruction que nous avons directement abordées :

- ▷ **Alignement de modèles 3D.** La tâches d'alignement de modèles 3D intervient dans le processus de reconstruction lorsque la stratégie utilisée implique la reconstruction de modèles 3D partiels à partir de sous-séquences, puis la fusion de ces modèles, nécessitant leur alignement. L'alignement est souvent effectué à l'aide de correspondances de points. Dans cette thèse, nous donnons des méthodes d'alignement basées sur des correspondances de droites. Notons que ces deux possibilités peuvent facilement être combinées. Nous pensons, même si cela nécessite des expérimentations supplémentaires, et notamment sur de longues séquences d'images, que l'utilisation conjointe de points et de droites dans les tâches d'alignement de modèles 3D permettrait d'améliorer la précision, comparé à l'utilisation exclusive de points ;
- ▷ **Ajustement de faisceaux et libertés de jauge.** Au cours de cette thèse, nous avons utilisé de nombreux algorithmes d'ajustement de faisceaux. Nous avons constaté que ces algorithmes étaient plus ou moins sensibles à la base de l'espace dans laquelle est exprimé le modèle 3D, ou de manière plus générale, la jauge. Une itération du même algorithme lancé sur les mêmes données exprimées dans deux jauges différentes peut conduire à des résultats très différents. Il existe à ce jour des méthodes pour fixer la jauge, mais peu d'entre elles sont basées sur des critères d'efficacité, et beaucoup se contentent d'utiliser une jauge triviale ou définie a priori d'une manière arbitraire. Nous pensons que l'efficacité des algorithmes d'ajustement de faisceaux peut être grandement améliorée par une bonne gestion de la jauge. Des résultats préliminaires publiés dans [15] le confirme.

Les commentaires suivants concernent deux tâches de reconstruction, la mise en correspondance et l'utilisation de textures, que nous n'avons abordées qu'indirectement. Ces deux tâches sont souvent aux deux extrémités d'un processus de reconstruction :

- ▷ **Mise en correspondance.** En dépit des nombreux travaux sur ce sujet, nous pensons que les algorithmes de mise en correspondance de primitives entre différentes images, notamment pour des images prises de points de vue éloignés, nécessitent d'être améliorés afin de devenir suffisamment fiables ;
- ▷ **Utilisation d'images de textures.** Le photoréalisme d'un modèle 3D reconstruit à partir d'images dépend entre autres de l'utilisation d'images de textures. Les modèles 3D que nous présentons au chapitre 5 et en annexe A reposent sur une extraction simpliste des textures, souvent à partir d'une seule image. Des idées telles l'utilisation d'une texture différente selon le point de vue du rendu ont emergées dans la littérature, mais peu de travaux étudient l'extraction de texture à partir de plusieurs images, car cela implique des problèmes tels la modélisation et l'estimation de la réflectance des surfaces. Nous pensons que ces problèmes sont ouverts et ont un impact important sur le photoréalisme des modèles 3D obtenus à partir d'images.

[1] Par exemple, la plupart des vecteurs ou matrices de coordonnées homogènes.

9.3 Problèmes ouverts

Le but de cette section est de discuter deux problèmes de vision par ordinateur que nous pensons ouverts : la reconstruction de scènes dynamiques et le rendu basé image contrôlable :

- **Reconstruction de scènes dynamiques.** Le monde qui nous entoure est dynamique, et des techniques de vision 3D par ordinateur permettant d'utiliser des images d'une scène dynamique ouvriraient un champ d'applications important, telles la reconstruction de modèles 3D dynamiques ou encore la vidéo surveillance autonome. Nous avons abordé le problème de la reconstruction de scènes dynamiques dans l'annexe B. On s'y reportera pour un état de l'art des méthodes de reconstruction dynamique. Nous pensons que les contributions à ce jour restent principalement au niveau de la recherche fondamentale, dans le sens où elles ne peuvent pas encore être appliquées à des problèmes réels. La reconstruction de scènes dynamiques est un domaine de recherche récent et nous pensons qu'il contient de nombreux problèmes ouverts ;
- **Rendu basé image contrôlable.** Cette thèse est centrée sur la reconstruction de modèles 3D à partir d'images. Une des applications les plus importantes de la reconstruction est la génération de vues virtuelles d'une scène réelle. Il existe une autre approche à ce problème, le rendu basé image. L'approche basée sur la reconstruction donne de bons résultats et a un aspect économique indéniable (le rendu de modèles 3D texturés est une des pierres d'angle de l'industrie des jeux vidéos), mais ses limites sont définies par la représentation paramétrique du modèle 3D reconstruit. La complexité de la scène pouvant être représentée est limitée : une scène très détaillée, par exemple des feuilles sur un arbre, nécessite un modèle 3D coûteux, en termes de reconstruction, de stockage et de manipulation. En partie pour ces raisons, une approche a récemment été développée, où les nouvelles images sont générées directement à partir des images originales. Ce processus, appelé le rendu basé image, a l'avantage que la représentation de la vue virtuelle est indépendante de la complexité de la scène. Cependant, les méthodes proposées ne permettent que le rendu de la scène acquise, c'est à dire qu'elles ne permettent aucun contrôle et aucune altération de la scène (par exemple déplacer un objet devant un autre), ce que permet naturellement l'utilisation d'un modèle 3D. Donc, d'un côté, les méthodes classiques permettent de modéliser des scènes de complexité réduite, mais offrent un niveau élevé de contrôle, et d'un autre côté, les techniques de rendu basé image gèrent des scènes complexes, mais ne confère que peu de souplesse au processus de rendu. Nous pensons qu'il y a un défi important à relever, en proposant une approche hybride, qui permettrait le rendu contrôlable de scènes complexes. Nous donnons quelques pistes dans cette direction. La première est l'utilisation combinée d'un modèle 3D grossier et du rendu basé image. L'approche serait de reconstruire une surface grossière et d'utiliser le rendu basé image pour la génération des images. Une deuxième piste est basée sur la composition d'un arrière-plan et d'un premier-plan, en utilisant par exemple le rendu basé image pour l'arrière-plan, et un modèle 3D pour le premier plan.

ANNEXE

A

DÉTECTION DE PLANS

Nous abordons le problème de la création automatique du modèle 3D d'un environnement humain, que nous représentons comme une collection de plans texturés.

Une approche typique est de reconstruire automatiquement un modèle 3D consistant de points. On indique ensuite manuellement leur appartenance à des plans. C'est la phase de segmentation en plans. Des images de texture sont ensuite extraites, éventuellement de manière conjointe à l'optimisation d'un critère d'erreur, par exemple l'erreur de reprojection.

Nous proposons une approche pour l'automatisation de la phase de segmentation en plans, c'est à dire permettant de détecter le nombre de plans et leurs paramètres à partir d'une reconstruction de points. Notre méthode est basée sur la génération d'hypothèses de plans par échantillonnage aléatoire de points. La pertinence d'une hypothèse est ensuite évaluée en estimant le contour du plan et en calculant une mesure de photoconsistance approchée.

Nous calculons finalement une estimation optimale de tous les paramètres de la scène, c'est à dire les plans, les points sur ces plans et les caméras. L'approche est validée sur des données simulées et réelles.

Ces travaux ont été publiés dans [10, 11].

A.1 Introduction

La reconstruction automatique de scènes rigides à partir d'images est un des problèmes clefs du domaine de la vision par ordinateur. De par son omniprésence dans les environnements humains, le cas des scènes planes par morceaux est spécialement important. La modélisation d'une scène par des primitives (points ou droites) et des plans a plusieurs avantages. D'une part, nous avons montré au chapitre 5 qu'elle permet d'obtenir une reconstruction de précision en général supérieure à celle obtenue par reconstruction individuelle de points. Elle confère au modèle 3D un aspect photoréaliste, par application d'une image de texture à chacun des plans. D'autre part, une collection de plans permet de modéliser la surface d'un grand nombre de scènes réelles, à un degré d'approximation raisonnable.

La plupart des systèmes de modélisation existants sont basés sur une reconstruction automatique de points ou droites, voir par exemple [32, 66, 160], suivie de deux étapes : la *segmentation* et l'*optimisation contrainte*. La segmentation consiste à définir des relations géométriques entre les points et les droites reconstruits, typiquement en donnant des formes géométriques de plus haut niveau (plans, cubes, sphères) [50, 117, 190] reposant sur les primitives reconstruites. Cette phase est souvent manuelle, un procédé laborieux dans le cas de scènes d'une géométrie complexe. Ce procédé est limité aux scènes constituées de formes géométriques prévues. L'étape d'optimisation contrainte consiste à optimiser la qualité du modèle 3D, en prenant en compte les contraintes géométriques précédemment définies. Cette approche permet de construire des modèles 3D hautement photoréalistes, mais requiert une intervention de plus en plus lourde de la part de l'utilisateur, lorsque la complexité de la scène augmente.

(a) (b)

FIG. A.1 – Exemple typique d'une scène plane par morceaux. (a) montre une image avec des points et leur segmentation en plans. (b) montre le modèle 3D texturé obtenu par notre algorithme.

Nous étudions le problème de l'automatisation de l'étape de segmentation. Notre étude concerne la *segmentation en plans*. La figure A.1 montre un exemple typique de scène plane par morceaux, sa segmentation automatique en plans et une vue du modèle 3D reconstruit. La segmentation automatique en plans permet, d'une part, d'enlever au processus de reconstruction la part d'intervention manuelle qu'il comporte, et d'autre part, d'augmenter le nombre de scènes pouvant être traitées, par rapport à une modélisation basée sur des formes plus complexes. Différents algorithmes ont été proposés dans la littérature [1, 9, 33, 54, 55, 61, 68, 113, 182, 201, 222]. Une approche souvent utilisée est basée sur l'utilisation récursive d'un estimateur robuste tel RANSAC [64] : le plan dominant, au sens d'un certain critère, est estimé, puis les points à l'extérieur de ce plan sont réinjectés dans l'estimateur robuste. Alors que ce schéma peut être performant dans le cas où les sous-ensembles segmentés doivent être disjoints, il s'avère peu efficace pour la détection de plans. La plupart des critères utilisés sont de nature purement géométrique. Ceci a plusieurs impacts sur le processus de segmentation, notamment la détection de faux plans et la non détection de certains plans, supportés par trop peu de points.

Ce chapitre est très fortement lié aux algorithmes de reconstruction plans par morceaux présentés au chapitre 5 : les algorithmes de détection de plans fournissent les données nécessaires à la reconstruction.

Contributions. Nous suivons le processus de modélisation exposé ci-dessus, consistant des étapes de reconstruction de points, segmentation en plans, puis reconstruction contrainte. Nous proposons une étape de segmentation automatique en plans, originale sur plusieurs aspects.

Notre algorithme est inspiré de RANSAC [64]. Il est basé sur l'échantillonnage aléatoire de points, permettant de créer des hypothèses de plans. Comme mentionné ci-dessus, une application itérative de ce schéma permet de détecter une série de plans, mais les ensembles de points coplanaires ainsi détectés sont disjoints. Nos contributions se situent au niveau d'une adaptation de ce schéma, permettant de détecter des ensembles de points non disjoints, c'est à dire qu'un point peut alors appartenir à plusieurs plans. Le critère standard utilisé pour juger la pertinence d'un plan hypothétique est purement géométrique (typiquement le nombre de points sur le plan) et entraîne la détection de faux plans, et rate certains vrais plans. Nous proposons un critère basé sur l'information photométrique induite par les images du plan hypothétique. Pour ce faire, nous proposons une procédure permettant de calculer le contour d'un plan, à partir de son équation et de ses images. La pertinence du plan est alors jugée à partir de sa photoconsistance, évaluée à l'aide d'un critère photométrique approximatif, la r-consistance proposée par Kutulakos [115, 116], permettant de s'affranchir d'une dépendance trop forte à la qualité de la reconstruction sous-jacente. De plus, ce critère permet de favoriser la détection des petits plans et pénalise les faux plans (les configurations géométriques approximativement coplanaires ne correspondant à aucun vrai plan). Notons que la méthode est applicable indifféremment du niveau de reconstruction – projectif, affine, métrique ou Euclidien.

Organisation de l'annexe. Nous donnons un état de l'art en §A.2. Nous proposons la méthode de segmentation automatique en plans en §A.3, où nous formulons notre méthode de segmentation en ensembles non disjoints, et le calcul de notre critère de pertinence photométrique. Finalement, nous donnons des résultats expérimentaux sur des données simulées et réelles en §A.4 et concluons et donnons des perspectives de travail en §A.5.

A.2 Etat de l'art

Cet état de l'art est centré principalement sur les travaux de segmentation automatique en plans. D'autres approches à la modélisation automatique à partir d'images sont la sculpture spatiale ("space carving" en Anglais) [116], la stéréo multi-vues, par exemple [115], et la triangulation consistante au niveau image [148].

Les travaux existants sur la segmentation automatique en plans peuvent être séparés en deux grandes classes, selon leur prise en compte de l'information photométrique. En particulier, des critères purement géométriques sont utilisés par Alon et Sclaroff [1] et Berthilsson et Heyden [33] pour segmenter une reconstruction éparse de points et par Faugeras et Lustman [61], Fornland et Schnörr [68] et Sinclair et Blake [182] en utilisant directement les points image sans reconstruction 3D explicite. Les inconvénients liés à l'utilisation de tels critères sont la détection de faux plans ainsi que la non détection des plans ne comportant que peu de points. Koch [113] propose une segmentation basée sur l'orientation de la surface d'une reconstruction dense de la scène. L'inconvénient de cette approche est sa forte dépendance à la surface reconstruite sous-jacente, difficile à obtenir de manière précise dans le cas général. Vestri et Devernay [222] utilisent une méthode robuste basée sur RANSAC pour segmenter la scène : ils détectent un grand nombre de plans par échantillonage aléatoire à partir d'une carte d'élévation numérique, puis fusionnent les plans trop proches.

L'information photométrique est prise en compte par Baillard et Zisserman [9], Dick et al. [54, 55] et Tarel et Vézien [201]. Les résultats obtenus par Baillard et Zisserman sont convaincants, mais au prix d'une complexité élevée. La méthode proposée par Dick et al. ne permet de modéliser les plans comportant peu de points qu'au prix de la connaissance a priori d'une collection de formes de type architectural. Pour la méthode de Tarel et Vézien, l'automatisme est au détriment de la qualité visuel du modèle obtenu. De plus la méthode n'est applicable qu'à une paire d'images.

modèle θ^m	$\mathcal{V}$	images de visibilité
	$\mathcal{B}_g$	support géométrique individuel
	$\mathcal{B}_p$	support photométrique dense
	∂	contour polygonal
paramètres θ^p	$\mathbf{\Pi}$	équation du plan
	$\mathcal{T}$	image de texture

TAB. A.1 – A l'issue de l'étape de segmentation, chaque plan $\boldsymbol{\Pi}$ détecté est modélisé par un modèle θ^m et ses paramètres θ^p. Ses deux ensembles de grandeurs sont déterminés par la phase de segmentation. La différence est que le modèle reste fixe par la suite, alors que les paramètres sont modifiés pour l'optimisation d'un certain critère.

Notons que la plupart de ces travaux, les méthodes de Fornland et Schnörr [68] et de Sinclair et Blake [182] exceptées, nécessite une reconstruction métrique de la scène, en d'autres termes, la connaissance ou l'estimation des données de calibrage des caméras.

A.3 Segmentation en plans

Nous décrivons l'étape de segmentation en plans. Nous proposons une procédure de segmentation par échantillonage aléatoire, basée sur une évaluation des hypothèses selon un critère photométrique. Nous donnons une description du processus et présentons la méthode de segmentation puis le calcul de l'évaluation photométrique d'un plan hypothètique.

A.3.1 Description du processus

La segmentation en plans consiste à affecter à chacun des points reconstruits un certain nombre de plans. Elle revient, de manière équivalente, à déterminer le nombre de plans z, et les points appartenant à chaque plan. Notons que les plans permettent d'effectuer la mise en correspondance des points image de manière plus fiable, car plus contrainte, qu'à l'étape initiale de reconstruction.

Nous définissons à ce stage la structure 3D comme un ensemble de plans, de points et de relations de multi-coplanarité. Nous séparons les grandeurs définissant la structure 3D en deux ensembles : le *modèle* et les *paramètres*. Nous définissons le *modèle* de la structure 3D comme des grandeurs déterminées par l'étape de segmentation, et non remises en question par l'étape ultérieure de reconstruction contrainte. Les *paramètres* de la structure 3D sont des grandeurs dont une valeur initiale est déterminée par l'étape de segmentation, et qui sont changées par l'étape de reconstruction contrainte, afin de minimiser un certain critère.

Le modèle et les paramètres de la structure 3D sont définis plus précisément dans le tableau A.1. Le modèle contient le nombre de plans z, et ce que nous appelons le modèle de chaque plan, comprenant les images de visibilité $\mathcal{V}$ (c'est à dire la liste des images dans lesquelles le plan est visible), le support géométrique individuel $\mathcal{B}_g$, c'est à dire la liste des points déterminés comme étant sur le plan, selon un critère géométrique, le support photométrique dense $\mathcal{B}_p$, déterminé selon un critère photométrique (voir ci-dessous), et le contour polygonal du plan, formulé comme une liste de points.

A.3.2 Segmentation par échantillonage aléatoire

Notre méthode de segmentation est basée sur l'itération d'estimations robustes par échantillonage aléatoire. L'idée est d'utiliser une méthode d'estimation robuste pour estimer le plan dominant. La segmentation des points exclus de ce plan est effectuée en les réinjectant dans l'estimateur robuste.

Nous examinons des méthodes d'estimation robustes et plus particulièrement la méthode RANSAC, puis son application à la segmentation de données en ensembles disjoints et non disjoints.

A.3.2.1 Méthodes d'estimation robustes

Les méthodes robustes sont utilisées en vision par ordinateur pour résoudre des problèmes telle l'estimation de la géométrie épipolaire [32, 208, 233] ou du tenseur trifocal [32, 210] en présence de données erronées. Les méthodes principalement utilisées sont les M-estimateurs, les moindres carrés médians [143] et RANSAC [64]. De tels estimateurs permettent d'estimer les paramètres d'un modèle donné a priori en présence de données erronées ou *outliers*. Ces dernières peuvent ensuite être distinguées des *inliers*, les données consistantes avec les paramètres du modèle. La principale différence entre ces estimateurs réside dans le taux maximal d'outliers pouvant être toléré.

Effectuer la segmentation de données à l'aide d'un estimateur robuste, comme esquissé précédemment, implique le choix d'un estimateur robuste pouvant tolérer un taux d'outliers élevé. Parmi les estimateurs cités ci-dessus, seul RANSAC tolère un taux d'outliers supérieur à 50%.

RANSAC. L'estimateur robuste RANSAC ("random sample concensus" en Anglais) cherche à trouver les paramètres d'un modèle pour lesquels le nombre d'inliers est maximal. Il fonctionne par échantillonage aléatoire. Un nombre minimal de données permettant l'estimation des paramètres du modèle est aléatoirement sélectionné, puis le nombre d'inliers est déterminé. Ce procédé est itéré puis les paramètres maximisant le nombre d'inliers sont sélectionnés.

Le nombre d'itérations est fonction d'un taux d'outliers supposé a priori, de la probabilité de réussite (en pratique on choisit une probabilité élevée, par exemple 99%) et du nombre de données minimal nécessaire à l'estimation des paramètres du modèle. En pratique, ce nombre d'itérations est reévalué dynamiquement à chaque sélection d'un nouveau jeu de paramètres maximisant le nombre d'inliers. Ceci permet de fixer un taux d'outliers a priori élevé sans augmenter le coût de l'algorithme. Une fonction indiquant la consistance d'une donnée avec les paramètres d'un modèle, servant à la discrimination des inliers, et donc à l'évaluation des hypothèses, est nécessaire à l'algorithme.

A.3.2.2 Segmentation en ensembles disjoints

L'algorithme RANSAC permet d'obtenir les paramètres dominants du modèle ainsi que ses outliers. Dans le cas de la segmentation, ces outliers ont deux interprétations possibles. Elles peuvent être de vrais données erronées ou bien des données consistantes avec d'autres paramètres du modèle. L'application successive de RANSAC sur les outliers permet donc de réaliser une segmentation des données.

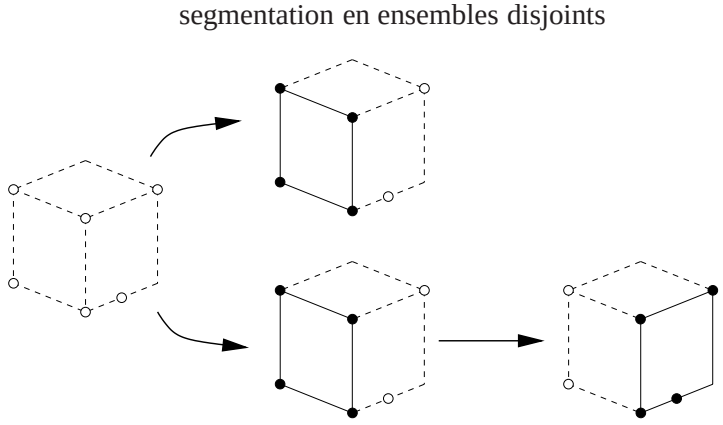


FIG. A.2 – Illustration de la différence entre un algorithme de segmentation en ensembles disjoints et un algorithme de segmentation en ensembles non disjoints. L'algorithme de segmentation en ensembles disjoints s'arrête après une itération car les points non segmentés (en blanc) ne permettent pas de trouver le deuxième plan. La segmentation en ensembles non disjoints ne s'arrête pas et trouve le deuxième plan.

Cette approche est performante dans le cas où les ensembles segmentés sont supposés être disjoints et a été employée avec succès dans le cas de la segmentation basée sur le mouvement, voir les travaux de Demirdjian et Horaud [51] par exemple, ou de groupements plans particuliers, voir les travaux de Schaffalitzky et Zisserman [170]. Dans le cas des plans, voir par exemple les travaux de Dick et al. [54], les sous-ensembles ne sont pas disjoints (un point peut être sur plusieurs plans) et la méthode ne permet de détecter que les plans principaux de la scène et de ne retrouver que des relations de coplanarité simple. La figure A.2 illustre les problèmes potentiels et la différence entre la segmentation en ensembles disjoints et non disjoints.

A.3.2.3 Segmentation en ensembles non disjoints

Une solution est de réaliser les applications successives de RANSAC sur l'ensemble du nuage de point. Par contre, afin de ne pas réestimer le même plan dominant à chaque itération, il faut vérifier lors de l'échantillonage aléatoire que le plan hypothétique n'a pas déjà été détecté. Nous le vérifions en comparant les plans déjà détectés et le plan hypothétique. Soient deux plans $\boldsymbol{\Pi}$ et $\boldsymbol{\Pi}'$ et leurs supports géométriques $\mathcal{B}_g$ et $\mathcal{B}'_g$. Ces deux plans sont déclarés identiques (noté $\boldsymbol{\Pi} \approx \boldsymbol{\Pi}'$) si le seuillage d'une mesure de distance $\mathcal{D}$ est positif :

$$\boldsymbol{\Pi} \approx \boldsymbol{\Pi}' \quad \Leftrightarrow \quad \mathcal{D}(\boldsymbol{\Pi}, \boldsymbol{\Pi}') < \gamma,$$

où γ est un seuil que nous définissons ci-dessous.

Notre mesure de distance doit avoir un sens physique. Par exemple, la comparaison des équations des deux plans de donnerais pas de bons résultats, car c'est une mesure purement algébrique. Nous choisissons de comparer les deux supports géométriques. En notant par le caractère $\#$ le cardinal d'un ensemble, notre mesure de distance s'écrit :

$$\mathcal{D}(\boldsymbol{\Pi}, \boldsymbol{\Pi}') \quad = \quad \frac{\#(\mathcal{B}_g \cap \mathcal{B}'_g)}{\#\mathcal{B}_g + \#\mathcal{B}'_g}.$$

Deux plans sont donc déclarés similaires lorsque le taux des inliers qui leur sont communs est supérieur à un seuil défini a priori. Dans nos expériences, nous avons utilisé un seuil variant entre 40 et 70 %, c'est à dire $0,4 < \gamma < 0,7$. Une autre possibilité de distance est de tester le support géométrique de chaque plan sur l'autre plan.

Notons que notre approche est différente de celle de Vestri et Devernay [222], qui détectent un grand nombre de plans, éventuellement très proches, puis appliquent une étape de fusion basée sur le support géométrique de chaque plan. L'application de RANSAC à la segmentation en plans nécessite un critère d'évaluation des hypothèses. Nous en proposons un dans la section suivante.

A.3.2.4 Evaluation d'un plan hypothétique

Un critère souvent utilisé pour juger un plan hypothétique, est le nombre de points sur le plan $\#\mathcal{B}_g$. C'est le critère de base de RANSAC. Un problème avec ce critère est dû au fait que certains points peuvent accidentellement se trouver sur un faux plan, ce qui arrive souvent en pratique. Le nombre de points appartenant à un plan hypothétique n'est donc pas un critère approprié pour évaluer la pertinence de cette hypothèse. La figure A.3 illustre ce problème et la différence avec un critère photométrique.

Nous évaluons la pertinence d'un plan hypothétique par la méthode suivante, illustrée sur la figure A.4. Nous disposons à ce stage de l'équation $\mathbf{\Pi}$ du plan et de son support géométrique $\mathcal{B}_g$. L'évaluation de la vraisemblance de l'hypothèse requiert de déterminer son modèle θ^m et ses paramètres θ^p. Nous proposons ci-dessous une méthode permettant de retrouver toutes ces grandeurs.

Images de visibilité $\mathcal{V}$. Nous notons $\mathcal{B}_p$ le support photométrique obtenu à l'issue de notre algorithme. Afin de déterminer les images de visibilité, nous supposons $\mathcal{B}_p \subseteq \mathcal{B}_g$, c'est à dire que tous les points du support photométrique sont géométriquement sur le plan. Cette hypothèse nous permet de déterminer $\mathcal{V}$ comme les images dans lesquelles l'ensemble des points de $\mathcal{B}_g$ est visible.

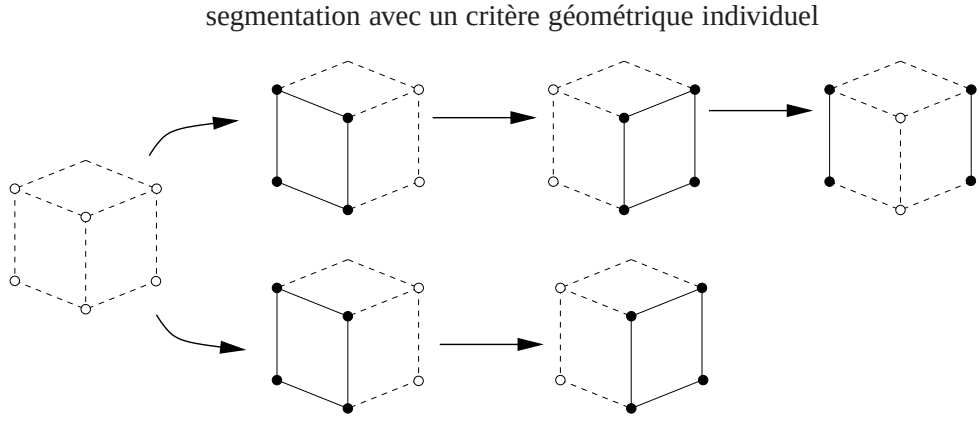


FIG. A.3 – Illustration de la différence de segmentation avec un critère géométrique individuel et un critère photométrique dense.

Contour polygonal ∂. La procédure suivie pour déterminer un contour polygonale ∂ du plan est illustrée sur la figure A.4. Ce contour polygonal s'appuie sur les points reconstruits. En notant $\mathrm{conv}(\mathcal{B}_g)$ l'enveloppe convexe des points du support géométrique $\mathcal{B}_g$, on peut écrire la relation suivante :

$$(\mathcal{B}_p \subseteq \mathcal{B}_g) \quad \Rightarrow \quad (\partial \subseteq \mathrm{conv}(\mathcal{B}_g)).$$

L'idée suivie est de former une triangulation des points du support géométrique, et d'en supprimer des triangles itérativement, selon un critère permettant de converger vers le contour recherché. Le critère utilisé pour valider ou invalider un triangle est basé sur sa photométrie, c'est à dire la comparaison de son apparence dans plusieurs images.

Notre algorithme est composé des étapes suivantes (voir la figure A.4). Pour commencer, nous alignons toutes les imagettes du plan (avec un contour $\mathrm{conv}(\mathcal{B}_g)$), et calculons la triangulation de Delaunay du support géométrique $\mathcal{B}_g$. L'alignement est effectué en utilisant des homographies de plan inter-images et l'interpolation bicubique des couleurs. Nous supposons que la triangulation ne coupe pas les arêtes du contour ∂, ce qui implique que l'invalidation de triangles dans $\mathrm{conv}(\mathcal{B}_g)$ doit converger vers ∂. Pour assurer une telle condition, nous utilisons une modification de la triangulation de Delaunay, permettant d'obtenir une triangulation consistante avec les images, c'est à dire minimisant une erreur photométrique sur les triangles. Nous renvoyons aux travaux de Morris et Kanade [148] pour plus de détails sur cette étape. La validité de chaque triangle est testée en utilisant le test de r-consistance, proposé par Kutulakos [115]. Nous supposons implicitement que la surface considérée est Lambertienne. Cette mesure présente une certaine robustesse à l'imprécision de la reconstruction et permet de gérer des surfaces approximativement planes. Le rayon de dispersion r, exprimé en pixels, contrôle l'erreur maximale d'un point reprojeté. Ce test détermine si les triangles alignés, provenant de différentes images, représentent la projection de la même surface plane. Le test est négatif si les couleurs de tous les pixels de tous les triangles ne sont pas présent dans les autres triangles, dans un disque de rayon r.

Cette évaluation permet de discriminer les plans, convexes ou non convexes, éventuellement troués, des fausses hypothèses et favorise la détection des petits plans, supportés par peu de points.

Les points à l'intérieur du contour ∂ forment le support photométrique $\mathcal{B}_p$ du plan. Le score du plan est donné par le nombre de triangles restant à la fin du processus.

A.4 Résultats expérimentaux

Nous décrivons des résultats expérimentaux obtenus sur des données simulées puis sur des données réelles.

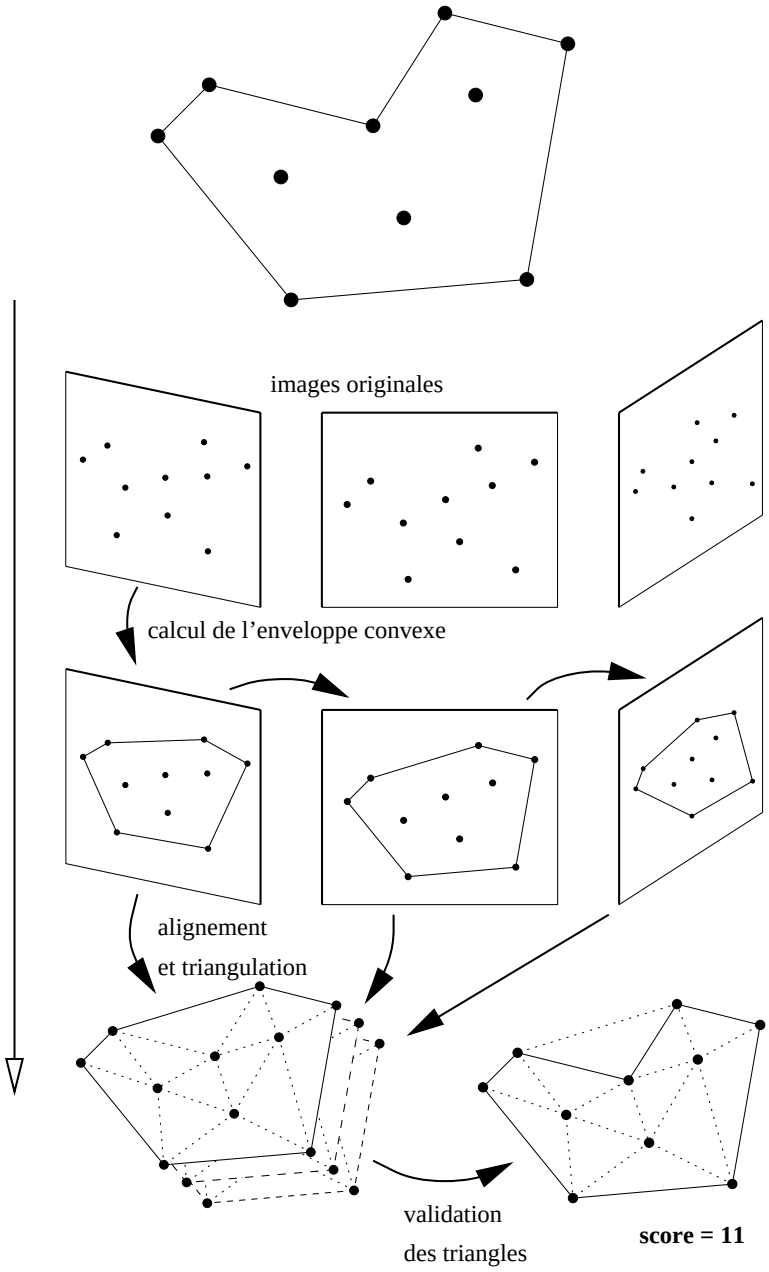
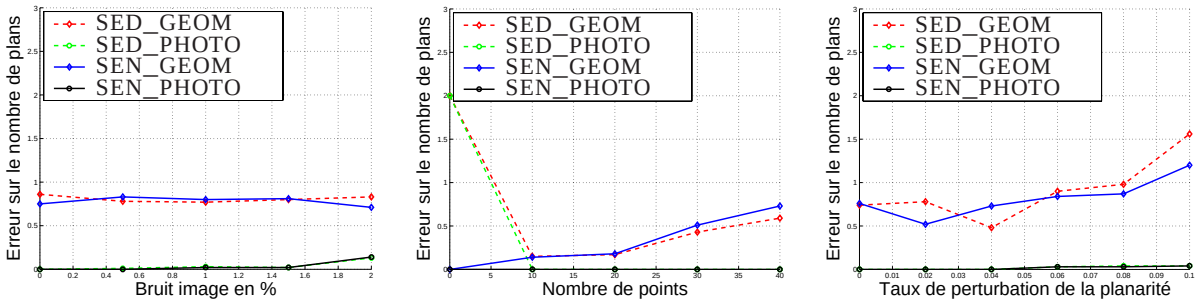


FIG. A.4 – Evaluation de la pertinence d'un plan hypothétique selon un critère photométrique dense.



(a) (b) (c)

FIG. A.5 – Comparaison de l'erreur dans le nombre de plans détectés pour différentes approches par rapport à différents paramètres de la scène simulée.

A.4.1 Données simulées

Nous comparons notre méthode de segmentation en plans à des méthodes existantes, consistant à utiliser une segmentation en ensembles disjoints [55] ou un critère purement géométrique [1, 182].

L'expérimentation consiste en un cube d'1 mètre de côté, observé par un jeu de caméras. Des points sont générés aléatoirement sur le cube, et éventuellement éloignés des plans les contenant afin de simuler une planarité non parfaite, puis projetés dans les images. Chaque face du cube est texturée à l'aide d'une image de texture aléatoire. La texture est de même projetée dans les images. Elle est normalisée de telle sorte que les valeurs des intensités soient réparties entre 0 et 1. Un nombre inférieur à 30, 5 et 1 points sont générés respectivement sur chaque face, arête et sommet du cube. Deux caméras d'une distance focal de 1000 pixels, séparées d'1 mètre et situées à 10 mètres du cube sont simulées, telles que 3 faces du cube soient visibles. Les paramètres intrinsèques des caméras ne sont pas utilisés, ce qui entraîne des reconstructions projectives.

Dans la suite, nous varions indépendamment certains paramètres de cette simulation pour comparer les différentes approches dans différentes conditions. Nous mesurons la qualité de la segmentation en utilisant la valeur absolue de la différence entre le nombre de plans simulés, c'est à dire 3, et le nombre de plans estimés. Nous mesurons des valeurs médianes sur 100 tirages. Le bruit appliqué sur la position des points image est 1 pixel. Le nuage de points 3D et les caméras sont obtenues par ajustement de faisceaux, voir le chapitre 3.

Nous comparons les méthodes de segmentation suivantes :

- ▷ SED_GEOM utilise une segmentation en ensembles disjoints, couplée à une évaluation purement géométrique des hypothèses : le cardinal du support géométrique, noté $\#\mathcal{B}_g$;
- ▷ SED_PHOTO est identique mais utilise une évaluation des hypothèses basée sur la vraisemblance photométrique, comme proposé en §A.3.2.4 ;
- ▷ SEN_GEOM utilise la segmentation en ensembles non disjoints présentée en §A.3.2.2 et le critère géométrique $\#\mathcal{B}_g$ pour l'évaluation des hypothèses ;
- ▷ SEN_PHOTO utilise la segmentation en ensembles non disjoints et le critère photométrique.

Nous décrivons les différentes situations expérimentales en variant un paramètre de la scène et les résultats obtenus :

- ▷ *bruit image ajouté*, figure A.5 (a) : un bruit Gaussien centré d'écart-type variant entre 0 et 0,02 (c'est à dire 2% du maximum) est ajouté à l'intensité de chaque pixel des images ;
- ▷ *nombre de points*, figure A.5 (b) : le nombre de points est varié entre 0 et 40 pour chaque face du cube. Nous utilisons aussi les coins du cube, mais aucun point sur ses arêtes ;
- ▷ *perturbation de la planarité*, figure A.5 (c) : les points 3D sont éloignés des plans les contenant avec des distances dont la distribution est centrée et uniforme, avec un écart-type variant entre 0 et 0,1 mètre.

Nous observons que dans le cas général, les méthodes *_PHOTO, basées sur l'information photométrique sont plus performantes que les méthodes *_GEOM basées sur un critère géométrique. En particulier, la méthode SEN_PHOTO donne de meilleurs résultats que les autres dans tous les cas.

Dans le cas d'un bruit image ajouté, nous observons que les méthodes *_GEOM sont moins efficaces que les

méthodes *_PHOTO. Ceci est dû au fait que les critères purement géométriques ne peuvent pas différencier les plans réels des configurations coplanaires accidentelles, ne correspondant à aucun plan réel. Comme le nombre de points est relativement petit, de telles configurations peuvent apparaître facilement.

Lorsque le nombre de points est varié, nous observons que pour de petits nombres (seuls les sommets du cube sont utilisés au début du graph), les méthodes SED_* sont moins performantes. Ceci est dû au fait que les plans ne peuvent être détectés que lorsque la multi-coplanarité (l'appartenance d'un point à plusieurs plans) est prise en compte. Lorsque le nombre de points augmente, les méthodes *_GEOM sont moins bonnes pour la même raison que celle donnée ci-dessus.

Dans le cas où la planarité de la scène est modifiée, les méthodes *_GEOM sont moins performantes, car la probabilité de configurations coplanaires ne correspondant à aucun plan augmente. Les performances des méthodes *_PHOTO sont peu affectées.

A.4.2 Données réelles

Nous donnons des résultats expérimentaux obtenus sur des images réelles. Nous examinons la séquence du modem puis la séquence "game system II". Il est plus facile de détecter les plans dans la première séquence : ces derniers sont bien distincts, et les deux images sont proches. Dans la deuxième séquence, les images sont plus éloignées, et les plans plus approximatifs.

A.4.2.1 Séquence du modem

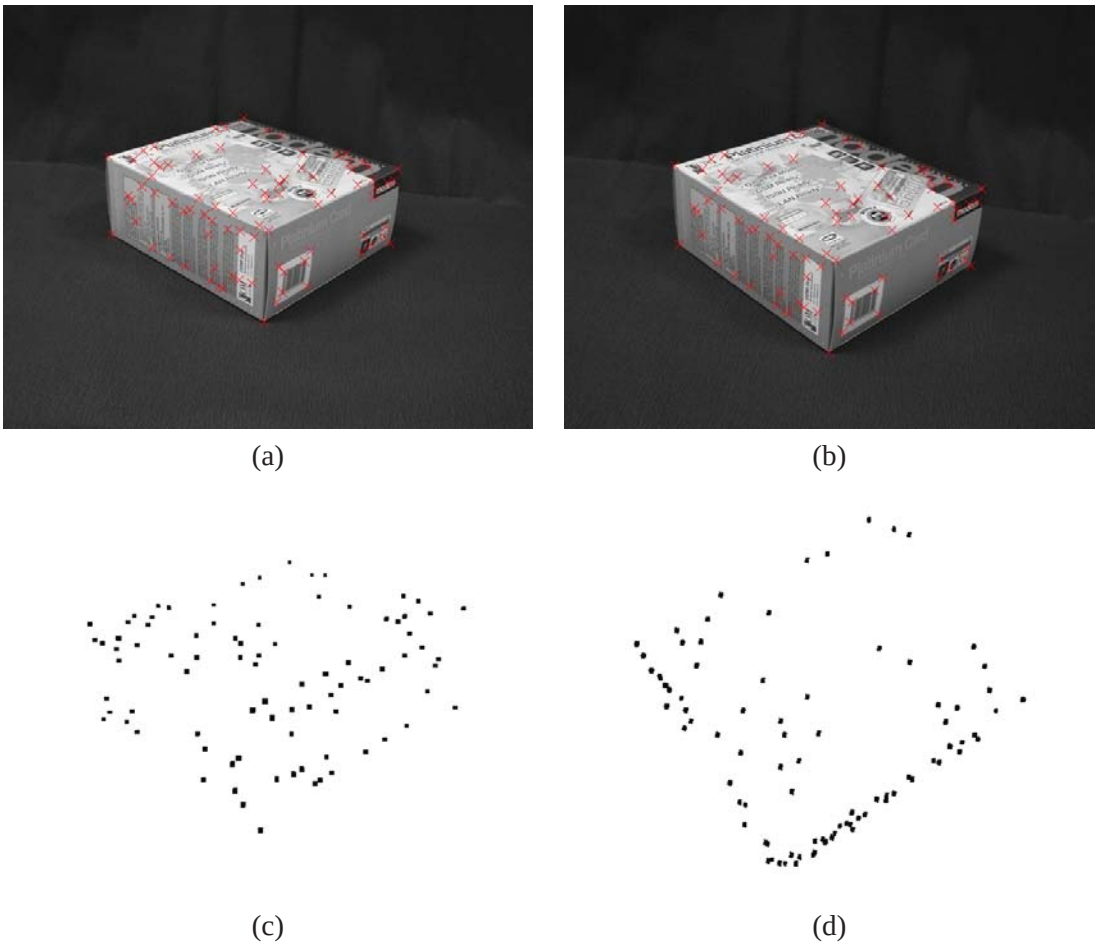
La séquence du modem consiste des deux images montrées sur la figure A.6. Le tableau A.2 donne l'erreur

FIG. A.6 – Les deux images de la séquence du modem.

de reprojection à différents stade de la modélisation.

Reconstruction de points et de caméras. Nous détectons des points d'intérêt à l'aide du détecteur de Harris et Stephen [80], visibles sur la figure A.7 (a) et (b), et utilisons une mise en correspondance couplée à l'estimation robuste de la géométrie épipolaire à l'aide de la méthode de Zhang [233]. Nous utilisons ensuite l'algorithme d'ajustement de faisceaux projectif proposé au chapitre 4, afin d'obtenir une reconstruction des points et des caméras. Nous utilisons le calibrage des caméras, obtenu de manière statique, afin de visualiser le modèle 3D obtenu en espace métrique, montré en figure A.7. Nous obtenons une erreur de reprojection initiale de 1,68 pixels, ramenée à 0,15 pixels par l'ajustement de faisceaux.

Segmentation en plans. L'algorithme détecte successivement trois plans, montrés en figure A.8. Les 3 plans principaux composant la scène sont détectés, et les points segmentés sur ces plans. On note que les points sur



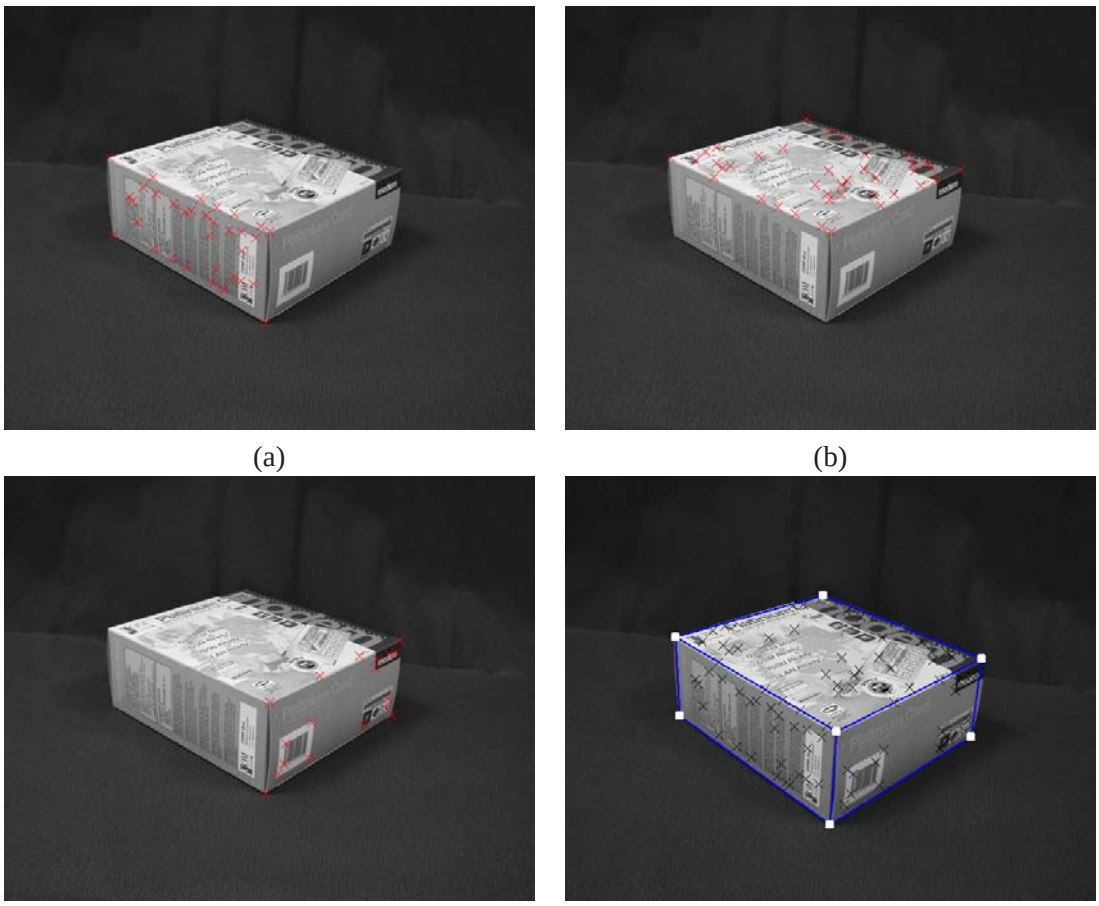
(a) (b)

(c) (d)

FIG. A.7 – Les points d'intérêt utilisés pour la séquence du modem (a) et (b), et la reconstruction obtenue, consistant de 90 points, vue de 3/4 (c) et de haut (d). On remarque, en particulier sur la vue (d), que la coplanarité des points est très approximative.

approche	étape	erreur de rep. (pixels)	# itérations
non contrainte	initialisation	1,68	-
	ajustement de faisceaux	0,15	5
contrainte	initialisation.	0,85	-
	ajustement de faisceaux	0,36	3

TAB. A.2 – Erreurs de reprojection obtenues à différentes étapes de l'algorithme et nombre d'itérations effectuées dans le cas des ajustement de faisceaux.



(a) (b)

(c) (d)

FIG. A.8 – La segmentation automatique en plans obtenue sur la reconstruction du modem : (a), (b) et (c) montrent les points appartenant aux 3 plans détectés, et (d) montre la reprojection du modèle 3D dans la première image.

(a) (b)

FIG. A.9 – Modèle obtenu pour le modem, (a) montre la reconstruction éparse contrainte par les plans et (b) montre le modèle 3D consistant de 3 plans délimités et des deux caméras. On observe en (a) que les points reconstruits sont parfaitement coplanaires.

FIG. A.10 – Rendus du modèle texturé du modem.

les arêtes sont correctement affectés à deux plans et le coin de la boîte aux trois plans. Nous avons utilisé un rayon de dispersion r=2 pixels. La segmentation est arrêtée lorsque l'algorithme ne peut détecter plus de deux triangles photoconsistants. Notons que le rayon de dispersion a peu d'influence sur le résultat de la segmentation car les plans sont bien distincts.

Reconstruction contrainte de plans, points et caméras. Le nuage de points est corrigé afin de satisfaire les contraintes géométrique de multi-coplanarité précédemment détectées. L'erreur de reprojection passe alors à 0,85 pixels. Afin d'obtenir un positionnement optimal, nous minimisons l'erreur de reprojection sous les contraintes de multi-coplanarité, à l'aide des techniques proposées au chapitre 5. L'erreur finale obtenue est 0,36 pixels, ce qui est très raisonnable. La reconstruction contrainte est montrée en figure A.9. Les textures sont extraites des images, et perspectivement corrigées, comme indiqué en §5.7.2.1, chapitre 5. La figure A.10 montre le modèle texturé ainsi obtenu.

Plus précisément, une image de texture est évaluée pour chaque plan en déformant chacune de ses images sur un plan fronto-parallèle, et en calculant la moyenne des couleurs sur les pixels de mêmes coordonnées. Ce calcul est justifié par l'hypothèse que la surface est Lambertienne.

A.4.2.2 Séquence "game system II"

La séquence "game system II" est constituée de 6 images, voir la figure A.11. Nous insistons sur le fait qu'il est difficile de détecter les plans sur cette séquence. En effet, les fenêtres sont éloignées de la façade

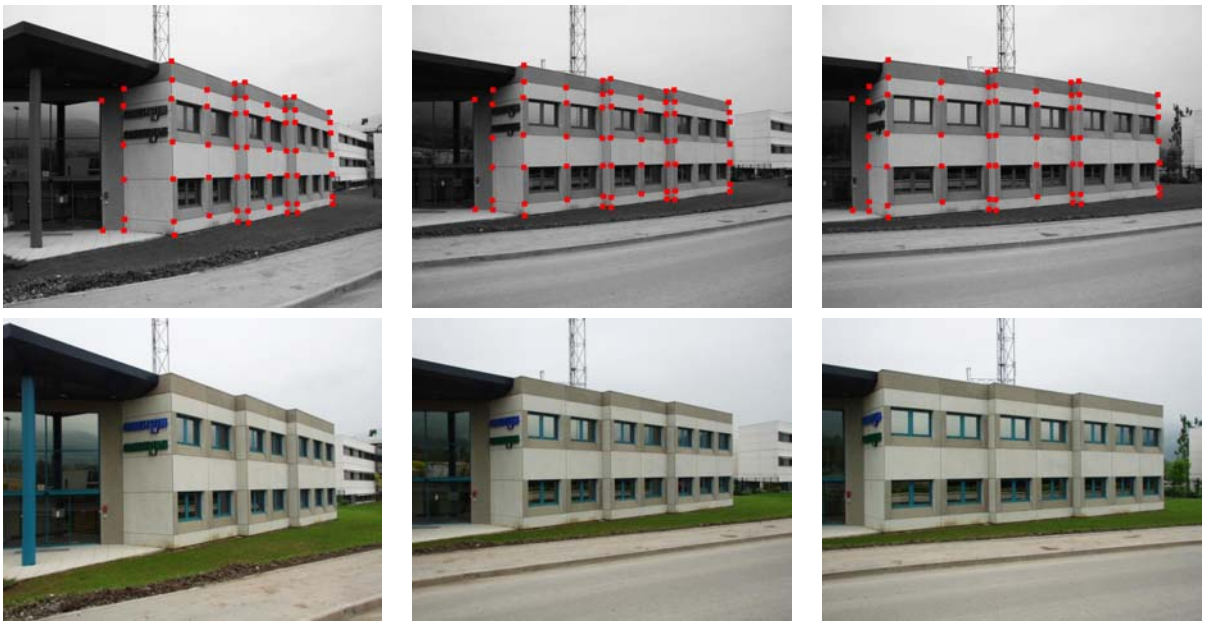


FIG. A.11 – La séquence game system II (3 des 6 images sont montrées), avec les correspondances de points utilisées. La deuxième ligne montre les images après correction de la distorsion radiale. On note une déviation significatives des fenêtres par rapport au plan principal de la façade.

principale, et ne peuvent pas être modélisées par des plans reposant sur les points reconstruits, car ces derniers sont pour la plupart sur la façade principale. De plus, l'apparence des vitres change entre les vues à cause d'effets spéculaires. Finalement, on note que certains plans sont très fins.

Reconstruction de caméras et de points. Nous donnons des correspondances de points manuellement sur les 6 images. Ces points sont visibles sur la figure A.11. Nous reconstruisons ces points et les 6 caméras, voir le chapitre 3. L'erreur de reprojection obtenue après initialisation est 3,82 pixels. Elle est réduite à 1,03 pixels par ajustement de faisceaux. Notons que l'estimation inclut un terme de distorsion radiale, que nous utilisons afin de corriger les images originales, voir la figure A.11. Le modèle 3D reconstruit est visible en figure A.12.

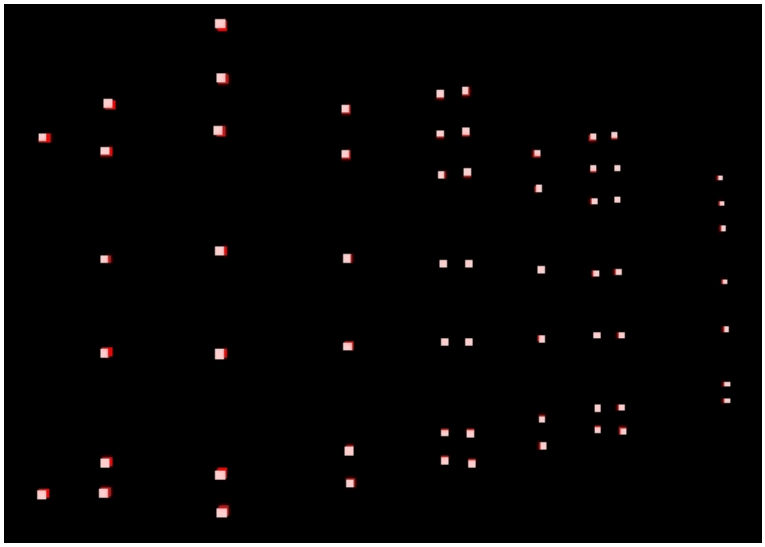
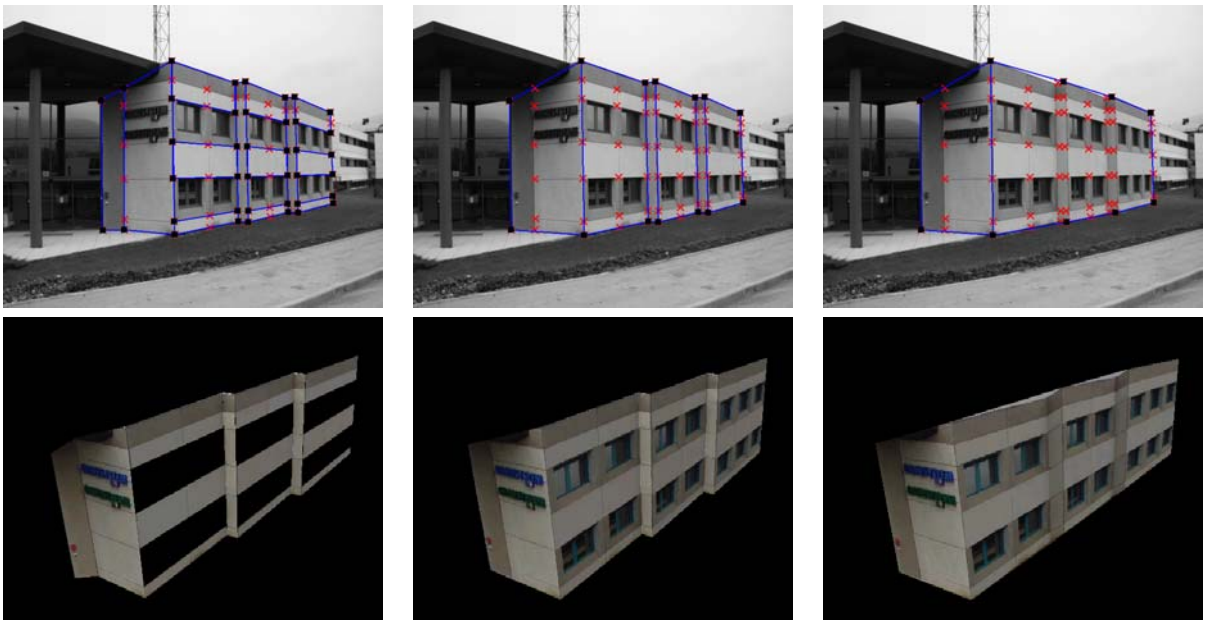


FIG. A.12 – Les points reconstruits après ajustement de faisceaux non contraint.

Segmentation en plans. Nous utilisons notre algorithme pour détecter les plans. Nous avons essayé différents rayons de dispersion. Le premier rayon de dispersion est $r = 1$ pixel. Il confère à l'évaluation des hypothèses un caractère strict. L'ensemble de plans retrouvés est très incomplet, en particulier, les plans incluant des fenêtres sont systématiquement non détectés, ce qui entraîne des "trous" dans le modèle 3D ainsi calculé, voir la figure A.13 (a).



(a) – $r = 1$ pixel (b) – $r = 5$ pixels (c) – $r = 10$ pixels

FIG. A.13 – Les plan détectés et le modèle 3D correspondant en utilisant successivement un rayon de dispersion $r = 1$ pixel, $r = 5$ pixels et $r = 10$ pixels, c'est à dire 0,2%, 1% et 2% de la taille de l'image. Dans le premier cas, (a), les plans modélisés sont précis, voir par exemple les deux plans à gauche. Les trous dans le modèle 3D correspondent aux fenêtres, qui ne peuvent ni être complètement modélisées, ni approchées par un plan avec un petit rayon de dispersion. Dans le deuxième cas, (b), la tolérance est plus élevée, donc les fenêtres peuvent être approchées par un plan. Notons que les deux plans à gauche ont été fusionnés. Pour le dernier cas, (c), deux plans suffisent à modéliser complètement la surface, ce qui entraîne plusieurs artefacts sur le rendu.

Nous avons ensuite essayé un rayon de dispersion $r = 10$ pixels, beaucoup plus important (2% de la taille de l'image). Ce rayon confère un aspect très tolérant à l'évaluation des plans hypothétiques. Le résultat est que les plans "proches" ont tendance à être agrégés. Le modèle 3D obtenu n'a plus de "trou", mais la modélisation est moins précise. Ce modèle est montré en figure A.13 (c).

Finalement, nous avons essayé un rayon de dispersion entre ces deux extrêmes, $r = 5$ pixels (1% de la taille de l'image). le modèle retrouvé est complet et a une apparence plus photoréaliste que dans les deux autres cas, voir la figure A.13 (b).

Cette expérience montre qu'avec un choix approprié pour le rayon de dispersion, notre méthode permet de détecter des surfaces approximativement planes et de petits plans.

Reconstruction de points, plans et caméras. Nous utilisons les techniques proposées au chapitre 5 afin de reconstruire les plans, les points et les caméras, sous les contraintes de multi-coplanarité détectés ci-dessus, avec $r = 5$ pixels. L'erreur de reprojection initiale est 4,32 pixels et passe à 1,58 pixels après ajustement de faisceaux plan par morceaux. Le modèle 3D calculé est montré est figure A.14. Les images de textures sont calculées à partir de toutes les images, sauf pour les fenêtres, qui sont extraites d'une seule image, afin d'éviter les effets de flou causés par le décalage par rapport à la façade principale et l'aspect spéculaire de la surface. Une autre possibilité pour le rendu de tels plans est d'utiliser une image de texture dépendante du point de vue ("view-dependent texture-mapping" en Anglais), comme proposé par Debevec et al. [50].

FIG. A.14 – Le modèle 3D texturé obtenu pour un rayon de dispersion $r = 5$ pixels.

rayon de dispersion r	σ_1	σ_2	μ
$r = 1$	0,0093	0,0284	0,0576
$r = 5$	0,0129	0,0370	0,0633
$r = 10$	0,0365	0,0511	0,1052

TAB. A.3 – Mesures métriques sur la reconstruction obtenues par ajustement de faisceaux plan par morceaux, avec les plans détectés pour différents rayons de dispersion (voir la figure A.13). Plus σ_1, σ_2 et μ sont petits, meilleure est la précision du modèle 3D.

Evaluation de la qualité. Nous avons effectué quelques mesures métriques sur la structure obtenue avec la reconstruction de points et de caméras, et celle obtenue après la reconstruction de points, de plans et de caméras, sous contraintes de multi-coplanarité, pour un rayon de dispersion $r = 1$ pixel. On se reportera au chapitre 5 pour des expériences similaires. Le but est ici un peu différent. On cherche à évaluer l'influence des contraintes de multi-coplanarité, mais aussi du rayon de dispersion, sur la précision de la reconstruction. Le tableau A.3 montre les résultats obtenus. Dans ce tableau, σ_1 et σ_2 sont les variances des longueurs des 6 arêtes verticales et horizontales respectivement, et μ est la moyenne de $|1 - \frac{2\alpha_i}{\pi}|$, où α_i est la mesure des angles droits. Nous avons effectué les mêmes mesures sur les modèles 3D obtenus avec les rayons de dispersion $r = 5$ pixels et $r = 10$ pixels. Ces mesures montrent que la précision de la reconstruction se dégrade lorsque le rayon de dispersion augmente. Ceci est dû au fait qu'un grand rayon de dispersion a tendance à rassembler et fusionner les plans (voir par exemple les deux plans à gauches sur la figure A.13 (a), fusionnés en A.13 (b) et (c)). Pour cette raison, la position individuelle de chaque plan est moins précise, et le modèle 3D de moins bonne qualité. En conséquence, il y a un compromis à trouver entre la qualité (précision) de la reconstruction et le fait que cette reconstruction modélise toute la surface de la scène.

A.5 Conclusions et perspectives

Conclusions. Nous avons présenté une méthode permettant de segmenter un nuage de points en plans. Cette méthode est utilisée dans un système de modélisation automatique pour scènes planes par morceaux.

La segmentation est basée sur une version modifiée de l'estimateur robuste RANSAC permettant de prendre

en compte le fait que les ensembles segmentés ne sont pas forcement disjoints (un points peut appartenir à plusieurs plans). La pertinence des hypothèses est évaluée à un niveau photométrique alors que ces mêmes hypothèses sont obtenue de façon géométrique. Ceci permet une meilleure qualité de segmentation, notamment par rapport à la détection de petits plans. La tolérance par rapport à un bruit dans la reconstruction est achevée par l'utilisation d'une mesure de photoconsistance approximative, la r-consistance.

Des expérimentations sur données simulées et réelles permettent de valider la méthode. Elles montrent que la méthode proposée donne de meilleurs résultats que les approches standards. Le rayon de dispersion r, permettant de détecter des configurations approximativement planes joue un rôle critique.

Perspectives. La méthode proposée donne de bons résultats lorsque le rayon de dispersion utilisé pour la r-consistance est approprié. La détermination automatique de ce rayon est un problème. On peut penser à utiliser des critères tels choisir le plus petit rayon de dispersion, tel que le modèle 3D retrouvé n'ait pas de trou.

Une autre perspective concernant la segmentation est de s'appuyer sur l'algorithme de segmentation générique proposé par Vidal et al. [223] (la décomposition en composantes principales généralisée). Cet algorithme s'applique directement au cas de la segmentation en plans, car la relation algébrique de coplanarité (appartenance d'un point à un plan) est bilinéaire. Cet algorithme fournirait donc une solution directe et simple au problème de la segmentation en plans, mais reposant sur un critère géométrique. Un des problèmes à résoudre avant d'appliquer cet algorithme serait celui des points n'appartenant à aucun plans.

Finalement, une perspective de travail intéressante serait d'adapter l'algorithme à différents types de primitives de plus haut niveau que les plans ainsi que l'incorporation d'un modèle de variation de la lumière, de type affine par exemple.

ANNEXE

B

GÉOMÉTRIE ET RECONSTRUCTION DYNAMIQUES – MOUVEMENTS LINÉAIRES, COPLANAIRES ET CONVERGENTS

Nous abordons le problème de la modélisation et de l'estimation de la géométrie d'une scène partiellement statique et partiellement dynamique. Plus particulièrement, nous considérons une caméra unique non calibrée observant une scène consistant de points statiques, et de points dynamiques, bougeant dans un plan de mouvement, le long de droites convergentes. Ce scénario décrit, par exemple, le mouvement de voitures sur une route, observées par une caméra non calibrée dont le mouvement est inconnu.

Le chapitre est divisé en deux parties : la modélisation et l'estimation de la géométrie dynamique, puis l'estimation de la géométrie statique standard, de manière consistante avec la géométrie dynamique.

Nous étudions les tenseurs d'appariement induits par la partie dynamique de la scène. La géométrie dynamique est décrite par ce que nous appelons le *tenseur-C7* et une collection de *tenseurs-C5*, dont la structure algébrique est proche de celle de la matrice fondamentale. Nous donnons des algorithmes d'estimation des paramètres de ce modèle.

Nous mettons en évidence le lien entre la reconstruction statique standard, et la modélisation dynamique. Ceci permet de reconstruire la position de chaque caméra avec un nombre restreint de points. Nous montrons comment le plan de mouvement (ou, de manière équivalente, l'homographie qu'il induit entre deux images), contenant seulement des points dynamiques, peut être reconstruit.

Nous donnons des résultats expérimentaux sur images réelles, en particulier sur une séquence de 60 images. Ces travaux ont été publiés dans [12].

B.1 Introduction

La plupart des travaux sur la géométrie multi-vues sont basés sur l'hypothèse que la scène observée est rigide ou statique. Cette contrainte permet de définir des tenseurs d'appariement entre deux, trois ou quatre images, par exemple, la matrice fondamentale. Ces tenseurs d'appariement encapsulent le mouvement et les paramètres intrinsèques des caméras ayant pris les images, et donc l'information géométrique nécessaire pour la reconstruction 3D. Les tenseurs d'appariement pour scènes statiques peuvent aussi être employés pour les scènes composées de multiples objets rigides en mouvements indépendants [45, 67, 224, 229], ce qui nécessite toutefois, qu'un nombre suffisamment important de primitives visuelles soit extrait pour chaque objet, rendant possible la segmentation, au moins implicitement. D'un autre côté, il y a une littérature de plus en plus abondante [7, 78, 79, 180, 197, 228, 230] examinant le cas de primitives en mouvements indépendants, souvent appelées des *primitives dynamiques*, par opposition aux *primitives statiques*. Le but de ces travaux est de fournir des algorithmes pour la *reconstruction dynamique*, c'est à dire la reconstruction des parties dynamiques de la scène, ainsi que des tenseurs d'appariement pour les images de primitives dynamiques. Des scénarios dynamiques très généraux et très contraints, impliquant des systèmes d'acquisition monoculaires ou stéréoscopiques, ont été examinés. Les motivations pour l'étude de la reconstruction dynamique sont essentiellement basées sur une volonté d'étendre le champ d'applications de la vision 3D par ordinateur. En effet, le monde réel est dynamique ; l'hypothèse de rigidité est donc très restrictive.

Nous considérons un scénario très contraint, formé d'une partie dynamique et d'une partie statique. La partie statique n'est pas contrainte, alors que, comme dans [7, 78, 79, 197, 228, 230], nous considérons que les points dynamiques bougent sur des *droites de mouvement*. Ce scénario est encore contraint en considérant que les droites de mouvement appartiennent à un *plan de mouvement* et convergent vers un *point d'incidence*. La figure B.2 illustre ce scénario. Notons qu'aucune hypothèse n'est faite sur le mouvement ou le calibrage de la caméra. Un exemple réel d'un tel scénario est le mouvement de points situés sur les toits de voitures, observées par une caméra de vidéo-surveillance en mouvement, voir la figure B.1.

FIG. B.1 – Un sous-ensemble des 60 images de la séquence vidéo de la route. La modélisation de la géométrie d'images fournies par une caméra en mouvement observant une route est une des applications du scénario considéré. Les difficultés pour la reconstruction 3D à partir d'une telle séquence proviennent du fait que la scène contient des primitives statiques et dynamiques, et qu'il n'y pas de contrainte sur le mouvement inconnu de la caméra.

Ce scénario s'inscrit dans des cas moins contraints précédemment examinés [7, 78, 79, 197, 228, 230]. Les algorithmes de reconstruction dynamique et les tenseurs d'appariement dynamiques correspondant peuvent donc être utilisés. Le principal problème est qu'ils nécessitent, en général, un nombre de correspondances de points élevé, rendant difficile l'utilisation, par exemple, d'un schéma d'estimation robuste par échantillonnage aléatoire tel RANSAC [64], souvent requis pour la conception d'un système automatique.

De plus, nous constatons les défauts spécifiques suivants. La méthode proposée par Avidan et Shashua [7]

vise à reconstruire des points en mouvement le long de droites, à partir d'images provenant d'une seule caméra. Elle requiert que le mouvement de la caméra soit connu, et des correspondances de points sur 5 vues. Les solutions proposées par Han et Kanade [78, 79] (basées sur une forme de factorisation non rigide) supposent que les primitives observées ait une vitesse constante, de même que la plupart des applications, excepté le tenseur de segmentation, proposées par Wolf et Shashua [228]. Les méthodes de Sturm [197] et Wolf et al. [230] cherchent à aligner des reconstructions 3D d'une scène dynamique, avec différentes hypothèses sur le mouvement des points. Par exemple, Sturm suppose que les points bougent dans des plans formant un pinceau. Ces méthodes impliquent l'utilisation de deux, ou plus, caméras synchronisées afin d'obtenir une vue 3D de la scène à chaque instant. Le "tenseur-H" proposé par Shashua et Wolf [180] modélise le mouvement de points statiques et de points dynamiques le long de droites dans un plan. Il nécessite au moins 3 images de points dynamiques et statiques pour pouvoir être estimé.

Contributions. Nous montrons que des tenseurs d'appariement et des algorithmes de reconstruction dynamique plus simples que ceux cités précédemment peuvent être formulés pour le scénario considéré.

Nous commençons par examiner le cas purement dynamique, c'est à dire lorsque seules des primitives dynamiques sont observées et que le mouvement de la caméra est inconnu. Nous montrons que dans le cas de deux vues, il existe un tenseur d'appariement dynamique, que nous appelons le *tenseur-C*, similaire à la matrice fondamentale. Plus précisément, nous appelons ce tenseur le *tenseur-C7*, afin de mettre en évidence ses 7 degrés de liberté. Les techniques d'estimation standards de la matrice fondamentale peuvent être appliquées pour estimer ce tenseur. Nous montrons que dans le cas multi-vues, l'information géométrique est encapsulée par un réseau de tenseurs-C, soumis à des contraintes de consistance. Nous proposons une modélisation basée sur un tenseur-C7 et des *tenseurs-C5*, une spécialisation du tenseur-C7. Un algorithme d'estimation consistante de cette géométrie est proposé.

Nous examinons ensuite le cas où des points dynamiques et statiques sont observés, c'est à dire que le scénario purement dynamique précédemment considéré est "plongé" dans une scène statique. Nous étudions le lien entre la modélisation dynamique précédemment considérée et l'estimation du mouvement projectif des caméras. Nous donnons des algorithmes pour l'estimation de ce mouvement contraint par les tenseurs-C.

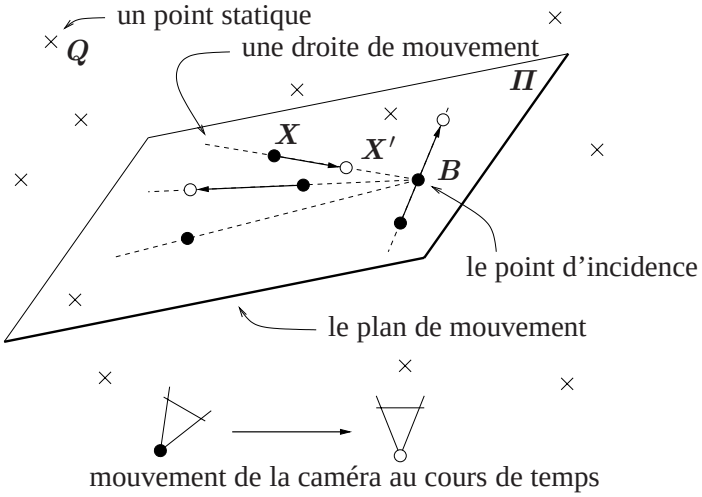


FIG. B.2 – Le scénario considéré consiste de points statiques non contraints, et de points dynamiques en mouvements indépendants sur un pinceau de *droites de mouvement*, convergants vers le *point d'incidence* et appartenant au *plan de mouvement*. Le plan de mouvement et le point d'incidence sont supposés statiques. Cependant, la plupart des résultats restent valides sans cette hypothèse. Dans la plupart des instances réelles de ce scénario, les droites de mouvement sont parallèles et le point d'incidence est donc à l'infini, voir les figures B.1 et B.3.

Organisation de l'annexe. Nous examinons la géométrie dynamique et son estimation en §B.2, où nous formulons les tenseurs-C. Nous nous intéressons ensuite en §B.3, aux contraintes exercées par ce tenseur sur la reconstruction des caméras, dans le cas où des points statiques et dynamiques sont observés. La section B.4 donne des résultats expérimentaux, puis la section B.5 conclut et donne des perspectives de travail. Le paragraphe suivant précise les notations spécifiques utilisées dans ce chapitre. La figure B.3 montre un exemple, la séquence des jouets, sur lequel nous illustrons notre modélisation et nos algorithmes au cours du chapitre.

FIG. B.3 – Les 3 images de la séquence des jouets, avec les points dynamiques utilisés. Notons que 4 points sont sur la voiture noire, qui effectue un dépassement dans la deuxième image, et ne satisfait donc pas le modèle dynamique associé aux autres points.

Notations. Les points observés sont désignés avec le caractère $\boldsymbol{U}$. Ces points peuvent être statiques ou dynamiques et sont alors notés $\boldsymbol{X}$ et $\boldsymbol{Q}$ respectivement. Les coordonnées successives au cours du temps sont respectivement notées $\mathbf{U}, \mathbf{U}', \mathbf{U}'', \ldots, \mathbf{X}, \mathbf{X}', \mathbf{X}'', \ldots$ et $\mathbf{Q}$. Les images de ces points sont désignées par $\mathbf{u}, \mathbf{u}', \mathbf{u}'', \ldots$, $\mathbf{x}, \mathbf{x}', \mathbf{x}''$ et $\mathbf{q}, \mathbf{q}', \mathbf{q}'', \ldots$. Le nombre de $'$ indique donc le numéro de la vue et l'instant correspondant. La figure B.2 illustre une partie de ces notations. Le point d'incidence est désigné par $\boldsymbol{B}$. Il est de coordonnées $\mathbf{B}$ et se projette en $\mathbf{b}, \mathbf{b}', \mathbf{b}'', \ldots$. Il est sur le plan de mouvement noté $\boldsymbol{\Pi}$.

B.2 La géométrie dynamique et son estimation

Cette section est centrée sur la modélisation de la partie dynamique du scénario considéré. Nous ne faisons aucune hypothèse sur le mouvement de la caméra, que nous supposons inconnu. Nous formulons des tenseurs d'appariement, pour le cas de deux vues puis pour le cas multi-vues.

B.2.1 Le cas de deux vues : le tenseur-C7

Nous étudions la formulation du *tenseur-C7*. Nous mettons en évidence une forte analogie avec la matrice fondamentale. Nous étudions ses propriétés et son estimation à partir de correspondances de points.

B.2.1.1 Formulation

Le tenseur-C7 encapsule le mouvement dynamique entre deux vues. Nous proposons une construction de ce tenseur. Il existe d'autres possibilités de construction, basées par exemple sur une matrice de projection $\mathbb{P}^3 \rightarrow \mathbb{P}^2$, dans l'idée du travail de Wolf et Shashua [228], ou sur la spécialisation d'un tenseur de liaison ("join tensor" en Anglais), voir le travail de Wolf et al. [230].

Soit $\mathsf{H}_{\boldsymbol{\Pi}}$ l'homographie 2D inconnue induite par le plan de mouvement, entre les deux vues considérées. Soit $\boldsymbol{X}$ un point dynamique et $\boldsymbol{x}$, $\boldsymbol{x}'$ ses projections dans les deux images. En utilisant $\mathsf{H}_{\boldsymbol{\Pi}}$, nous pouvons prédire la projection $\tilde{\boldsymbol{x}}'$ de $\boldsymbol{X}$ dans la deuxième image, comme si $\boldsymbol{X}$ était statique : $\tilde{\mathbf{x}}' \sim \mathsf{H}_{\boldsymbol{\Pi}}\mathbf{x}$. La droite image $\boldsymbol{m}'$, projection de la droite de mouvement associée à $\boldsymbol{X}$ dans la deuxième vue est donnée en joignant $\tilde{\boldsymbol{x}}'$ et la projection du point d'incidence $\boldsymbol{b}'$:

$$\mathbf{m}' \quad \sim \quad \mathbf{b}' \wedge \tilde{\mathbf{x}}' \quad \sim \quad [\mathbf{b}']_{\wedge}\mathsf{H}_{\boldsymbol{\Pi}}\mathbf{x}.$$

Il est évident qu'une condition nécessaire pour que $\boldsymbol{x}$ et $\boldsymbol{x}'$ soient des instances du même point 3D dynamique, est que $\boldsymbol{x}'$ soit sur la droite $\boldsymbol{m}'$, d'où l'équation :

$$\mathbf{x}'^{\mathsf{T}}\mathsf{C}\mathbf{x} \;=\; 0 \quad \text{avec} \quad \mathsf{C} \;\sim\; [\mathbf{b}']_{\wedge}\mathsf{H}_{\boldsymbol{\Pi}}, \tag{B.1}$$

où C est le *tenseur-C7* (a 7 degrés de liberté, voir ci-dessous). Ce tenseur encapsule la signature image de la géométrie dynamique pour deux vues.

B.2.1.2 Analogie avec la matrice fondamentale

Un aspect frappant du raisonnement précédent est la très forte analogie pouvant être faite entre le tenseur-C7 et la matrice fondamentale. Plus précisémment, on peut associer les projections du point d'incidence aux épipôles. L'homographie $\mathsf{H}_{\boldsymbol{\Pi}}$ peut être associée à une homographie de plan. Elle correspond réellement à une homographie de plan si et seulement si le plan de mouvement est statique. L'homographie 1D reliant les pinceaux de droites de mouvement, appelée la *transformation des droites de mouvement*, correspond à la transformation épipolaire. Cette analogie nous permet d'utiliser, pour le tenseur-C7, le formalisme et les résultats donnés au chapitre 4, concernant la géométrie épipolaire. Notons que lorsque les points dynamiques ne bougent pas entre les deux vues, le point d'incidence n'est pas défini. Par l'analogie décrite ci-dessus, ceci correspond à l'observation d'un plan statique par deux caméras : dans ce cas, la position des épipôles n'est pas définie.

B.2.1.3 Propriétés

Grâce à l'analogie proposée ci-dessus, plusieurs propriétés du tenseur-C7 peuvent être facilement formulées. Le tenseur-C7 est de rang 2 et a 7 degrés de liberté. Les coordonnées de la projection du point d'incidence dans la première image, respectivement dans la deuxième, sont données par l'espace nul droit, respectivement gauche, du tenseur-C7 : $\mathsf{C}\mathbf{b} = \mathsf{C}^{\mathsf{T}}\mathbf{b}' = \mathbf{0}$.

La matrice fondamentale définit la famille des homographies de plan de dimension 3, voir le chapitre 2, §2.11. Cette famille s'exprime à partir d'une homographie de plan canonique. De manière similaire, on peut définir une famille de dimension 3 d'homographies 2D compatibles avec le tenseur-C7. Une homographie canonique, notée $\check{\mathsf{W}}$, est formée à partir de C, et la famille de dimension 3 des homographies 2D compatibles avec C est paramétrée par le vecteur $\mathbf{a} \in \mathbb{R}^3$ comme :

$$\check{\mathsf{W}} \;\sim\; [\mathbf{b}']_{\wedge}\mathsf{C} \quad \text{et} \quad \mathsf{W}_{\mathbf{a}} \;\sim\; \check{\mathsf{W}} + \mathbf{b}'\mathbf{a}^{\mathsf{T}}. \tag{B.2}$$

Notons que la famille d'homographies 2D $\mathsf{W}_{\mathbf{a}}$ contient l'homographie de plan $\mathsf{H}_{\boldsymbol{\Pi}}$ induite par le plan de mouvement. Nous verrons en §B.3.1 que lorsque la matrice fondamentale F est connue, il est possible de retrouver $\mathsf{H}_{\boldsymbol{\Pi}}$, en calculant l'intersection des familles formées par les $\mathsf{W}_{\mathbf{a}}$ et les homographies planes.

Soit D une homographie 2D pour droites, c'est à dire duale à une homographie $\mathsf{W}_{\mathbf{a}}$, nous avons la relation : De manière duale, on obtient les expressions :

$$\mathsf{C} \;\sim\; \mathsf{D}[\mathbf{b}]_{\wedge}, \tag{B.3}$$

Pour comprendre cette expression, considérons un point dynamique $\boldsymbol{x}$ dans la première image : $[\mathbf{b}]_{\wedge}\mathbf{x}$ est l'équation de la droite de mouvement associée à $\boldsymbol{x}$. En conséquence, D étant une homographie pour droites, $\mathsf{D}[\mathbf{b}]_{\wedge}\mathbf{x}$ est la droite de mouvement associée à $\boldsymbol{x}$, dans la deuxième image. La matrice D encapsule la transformation des droites de mouvement et la position du deuxième point d'incidence $\boldsymbol{b}'$.

B.2.1.4 Estimation

Une autre conséquence de l'analogie entre le tenseur-C7 et la matrice fondamentale est qu'il est possible d'appliquer n'importe quel algorithme d'estimation de la matrice fondamentale à partir de correspondances de points, au calcul du tenseur-C7. Nous utilisons l'algorithme des huit points [86], avec un estimateur robuste tel RANSAC [64, 233] afin de calculer une solution initiale pour le tenseur-C7 et de détecter les faux appariements et les éventuels points dynamiques ne satisfaisant pas le modèle (par exemple, les voitures en dépassement).

Nous utilisons ensuite un algorithme d'ajustement de faisceaux pour calculer une solution optimale du tenseur-C7, voir le chapitre 4 pour plus de détails. Dans ce cas, les profondeurs projectives des points représentent leur déplacement le long des droites de mouvement. La figure B.4 montre les droites de mouvement calculées par cette méthode sur les images de la séquence des jouets. Notons que les points sur la voiture noire, effectuant un dépassement, ont été détectés et éliminés. Sur ces trois images, deux tenseurs-C ont été estimés, entre les images 1 et 2 et entre les images 2 et 3. Les droites de mouvement sont donc définies deux fois dans l'image 2. On observe qu'elles ne coïncident pas.

FIG. B.4 – Les droites de mouvement estimées par ajustement de faisceaux sur des paires d'images : en blanc entre les images 1 et 2, et en noir en les images 2 et 3. Notons que l'estimation robuste permet de détecter et d'éliminer les points sur la voiture noire effectuant un dépassement. Pour l'image 2 (au milieu), les droites de mouvement sont définies par deux tenseurs-C7.

B.2.2 Le cas multi-vues : le tenseur-C5

Nous montrons que les relations géométriques entre n vues de points dynamiques sont contenues dans $n-1$ tenseurs-C, soumis à des contraintes de consistance, exprimant le fait que le point d'incidence est unique. Ces contraintes sont modélisées par une spécialisation du tenseur-C7 en un *tenseur-C5*.

On note que l'analogie entre la matrice fondamentale et le tenseur-C7 dans le cas de deux vues, ne se poursuit pas dans le cas multi-vues. En effet, la géométrie d'un système de n caméras non calibrées a $11n-15$ degrés de liberté, alors que la géométrie dynamique se révèle en avoir $5n-3$ (voir ci-dessous). De plus, un point 3D statique a 3 degrés de liberté, alors qu'un point dynamique en a $n+1$ (sous réserve qu'il soit visible dans les n images, voir ci-dessous).

B.2.2.1 Analyse des degrés de liberté et formulation

D'après les résultats de la section précédente, nous savons que pour $n=2$ vues, la géométrie dynamique a 7 degrés de liberté. Elle est représentée par un tenseur C7 (a 7 degrés de liberté), noté C. Considérons une troisième vue de la même scène, ayant des points dynamiques en commun avec au moins une des deux premières vues, par exemple la deuxième. On peut calculer un tenseur-C C' entre la deuxième et la troisième vue, comme sur l'exemple de la figure B.4. Cependant, il faut garder en mémoire qu'à un instant donné, le point d'incidence $\boldsymbol{B}$ a une position unique. Par conséquent, C et C' doivent donner la même position à sa deuxième image $\boldsymbol{b}'$, pour être consistants. Ceci exerce deux contraintes sur la géométrie, et laisse donc $7+7-2=12$ degrés de liberté pour la géométrie dynamique de 3 vues. Il est trivial de pousser le raisonnement pour n vues, et de voir que cette dernière a alors $7+5\times(n-1)=5n-3$ degrés de libertés.

La géométrie dynamique d'un ensemble de n images peut donc être représentée par un tenseur-C7 entre deux vues de référence et un réseau de $n-2$ *tenseurs-C5*. Un tenseur-C5 est un tenseur-C7 avec un point d'incidence contraint. Nous donnons un algorithme pour l'estimation d'un tenseur-C5 dans la section suivante.

De manière équivalente, on peut voir la géométrie dynamique de n caméras comme n images du point d'incidence, et $n-1$ transformations des droites de mouvement. Cette représentation a l'avantage de ne pas

utiliser des vues de référence. En pratique, on utilise cette modélisation pour l'estimation non-linéaire. La modélisation basée sur les tenseurs-C est utilisée pour l'estimation linéaire.

Concernant la structure dynamique, le nombre de degrés de liberté de chaque point dépend du nombre d'images dans lesquelles il est visible. Un point visible dans toutes les images a $2 + (n-1) = n+1$ degrés de liberté, correspondant à sa position dans le plan de mouvement a un instant particulier, et a $n-1$ positions successives le long de sa droite de mouvement.

B.2.2.2 Estimation linéaire par succession de tenseurs-C

Supposons qu'un tenseur-C7 ait était calculé entre deux vues particulières. Les tenseurs-C5 doivent être calculés étant donnée une projection du point d'incidence, c'est à dire sous la contrainte que leur noyau gauche ou droit est connu. Cette contrainte est importante car elle permet de rendre l'estimate consistante, et de n'utiliser que 5 correspondances de points, au lieu de 7. Nous proposons un algorithme d'estimation linéaire.

L'algorithme global d'estimation consiste à estimer un tenseur-C7 entre deux images, puis un tenseur-C5 pour chaque image supplémentaire. Il peut donc être utilisé à la volée : chaque nouvelle image est ajoutée au système par l'estimation d'un tenseur-C5.

Notre algorithme linéaire est basé sur le formalisme développé au chapitre 4, et sur une idée originale de Hartley [83]. On suppose que le point d'incidence dans la première image, $\boldsymbol{b}$, est donné. Partons de l'équation (B.1), et remplaçons le tenseur-C par sa factorisation partielle en homographie duale+point d'incidence, donnée par l'équation (B.3) :

$$\mathbf{x'}^{\mathsf{T}} \mathsf{D} [\mathbf{b}]_{\wedge} \mathbf{x} \quad = \quad 0, \tag{B.4}$$

où la matrice D contient la transformation des droites de mouvement et la position du point d'incidence $\boldsymbol{b'}$ dans la deuxième image. Le but est de calculer D à partir de cette équation. Pour ce faire, considérons la décomposition en valeurs singulières suivante :

$$([\mathbf{b}]_{\wedge})_{(3\times3)} \quad = \quad \mathsf{U}_{(3\times3)} \mathsf{\Sigma}_{(3\times3)} \mathsf{V}^{\mathsf{T}}_{(3\times3)},$$

où $\mathsf{\Sigma} = \mathrm{diag}(\sigma_1, \sigma_2, 0)$. Une solution efficace pour le calcul de cette décomposition est donnée dans [83]. En substituant cette décomposition dans l'équation (B.4), nous obtenons la contrainte réduite suivante :

$$\mathbf{x'}^{\mathsf{T}} \bar{\mathsf{D}} \bar{\mathbf{y}} \quad = \quad 0,$$

avec :

$$\begin{pmatrix} & & 0 \\ \bar{\mathsf{D}}_{(3\times2)} & & 0 \\ & & 0 \end{pmatrix} \quad = \quad \mathsf{D}\mathsf{U}\mathsf{\Sigma} \quad \text{et} \quad \begin{pmatrix} \bar{\mathbf{y}}_{(2\times1)} \\ y \end{pmatrix} \quad = \quad \mathsf{V}^{\mathsf{T}}\mathbf{x}.$$

Notons que $\bar{\mathsf{D}}$ est définie par un vecteur-6 de coordonnées homogènes, noté $\bar{\mathbf{d}} \in \mathbb{P}^5$, la vectorisation par ligne de $\bar{\mathsf{D}}$, ce qui est consistant avec le fait que le tenseur-C5 ait 5 degrés de liberté. En développant l'équation de la contrainte réduite, nous obtenons le système linéaire suivant pour $\bar{\mathbf{d}}$:

$$\mathsf{A}_{(m\times6)} \bar{\mathbf{d}}_{(6\times1)} \quad = \quad \mathbf{0}_{(m\times1)} \quad \text{avec} \quad \mathsf{A} \quad = \quad \begin{pmatrix} & & \cdots & & & \\ x'_1\bar{y}_1 & x'_1\bar{y}_2 & x'_2\bar{y}_1 & x'_2\bar{y}_2 & \bar{y}_1 & \bar{y}_2 \\ & & \cdots & & & \end{pmatrix},$$

où m est le nombre de correspondances de points dynamiques. Notons que ce système a en général une solution unique pour $m = 5$, et peut être résolu au sens des moindres carrés pour $m > 5$. A partir de $\bar{\mathbf{d}}$, on forme $\bar{\mathsf{D}}$, puis le tenseur-C5 comme :

$$\mathsf{C} \quad \sim \quad \begin{pmatrix} & & 0 \\ \bar{\mathsf{D}}_{(3\times2)} & & 0 \\ & & 0 \end{pmatrix} \bar{\mathsf{V}}^{\mathsf{T}}.$$

La figure B.5 montre le résultat obtenu avec cet algorithme sur la séquence des jouets. Notons que cette méthode minimise une fonction de coût basée sur la distance algébrique entre les points et les droites de mouvement. Une fonction de coût non-linéaire, par exemple l'erreur de reprojection ou la distance Euclidienne entre les points et les droites de mouvement, peut être minimisée de manière consistante avec un algorithme d'optimisation non-linéaire, utilisant une paramétrisation de la matrice fondamentale permettant de fixer la position d'un épipôle (par analogie, un point d'incidence), par exemple la représentation orthonormale ou la paramétrisation de Zhang, voir le chapitre 4.

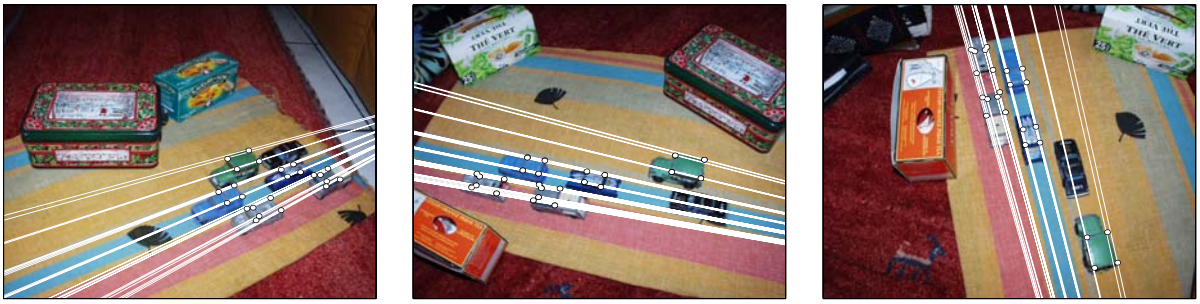


FIG. B.5 – Les droites de mouvement estimées de manière consistante. On note qu'elles coïncident parfaitement dans la deuxième image.

B.2.2.3 Estimation non-linéaire globale

La section précédente décrit une méthode d'estimation linéaire de la géométrie dynamique de n vues. Le résultat est sous-optimal, dans le sens où les données ne sont pas toutes prises en compte de manière égale. Dans cette section, nous proposons un algorithme optimal. Dans le cas de deux vues, nous utilisons un ajustement de faisceaux classique, ce qui est rendu possible par l'analogie entre la matrice fondamentale et les tenseurs-C. Dans le cas multi-vues, il n'existe plus d'analogie entre le réseau de tenseurs-C et la géométrie d'un système de caméras. Un système d'estimation spécifique est nécessaire.

Comme pour un ajustement de faisceau classique, l'idée est d'estimer des points corrigés les plus proches possibles des points mesurés, et satisfaisant un modèle géométrique consistant. Nous proposons la paramétrisation consistante suivante. Le mouvement dynamique a $5n - 3$ degrés de liberté : nous le paramétrons par n images du point d'incidence, et $n - 1$ homographies de $\mathbb{P}^1$ (a 3 degrés de liberté), les transformations des droites de mouvement, ce qui donne les $2n + 3(n - 1) = 5n - 3$ degrés de liberté requis. Chaque point dynamique peut être paramétré par un point image et $n - 1$ points image contraints sur les droites de mouvement correspondantes, ce qui donne les $2 + (n - 1) = n + 1$ degrés de liberté requis. La paramétrisation d'un point image sur une droite image donnée avec un paramètre est possible, comme décrit au chapitre 5, §5.4.1.5.

B.3 Estimation consistante des géométries dynamiques et statiques

Dans cette section, nous considérons que des points dynamiques et statiques sont observés conjointement. Le but est d'estimer la géométrie statique (la position des caméras) en prenant en compte les contraintes apportées par la géométrie dynamique. Nous mettons en évidence le lien entre la géométrie dynamique, représentée par un réseau de tenseurs-C, et la géométrie statique standard, représentée par une matrice fondamentale et des matrices de projection. Nous montrons comment contraindre la reconstruction des caméras et des points statique par la partie dynamique de la scène, en d'autres termes, qu'elle est l'influence des tenseurs-C sur la matrice fondamentale et les matrices de projection. Nous montrons que l'intégralité de la scène peut être reconstruite (position des caméras, points statiques et dynamique à chaque instant, plan de mouvement, point d'incidence).

B.3.1 Le cas de deux vues

La géométrie statique de deux vues non calibrées est représentée par la matrice fondamentale F. Nous examinons le lien entre F et le tenseur-C7 C. Nous montrons que si ces deux entités sont connues, alors l'homographie $\mathsf{H}_{\boldsymbol{\Pi}}$ induite par le plan de mouvement entre les deux vues peut être déterminée. Nous donnons une solution analytique pour cette homographie, en termes de F et de C, et établissons le lien avec les algorithmes standards d'estimation d'une homographie.

B.3.1.1 Le lien entre C et F

Le lien entre la matrice fondamentale et le tenseur-C7 implique indirectement l'homographie $\mathsf{H}_{\boldsymbol{\Pi}}$ induite par le plan de mouvement. La matrice fondamentale définit la famille des homographies de plan de dimension 3, et réciproquement, toute homographie de plan contraint la matrice fondamentale comme :

$$\mathsf{F} \quad \sim \quad [\mathbf{e}']_\wedge \mathsf{H}_{\boldsymbol{\Pi}}. \tag{B.5}$$

De manière analogue, il existe une famille d'homographies de dimension 3 compatibles avec le tenseur-C, paramétrée par l'équation (B.2) :

$$\mathsf{W}_{\mathbf{a}} \quad \sim \quad [\mathbf{b}']_\wedge \mathsf{C} + \mathbf{b}'\mathbf{a}^\mathsf{T}.$$

Nous savons que les deux familles d'homographies compatibles avec C et F ont un élément en commun, qui est l'homographie induite par le plan de mouvement $\mathsf{H}_{\boldsymbol{\Pi}}$. Nous pouvons donc écrire :

$$\exists \mathbf{a} \in \mathbb{R}^3 \quad | \quad \mathsf{H}_{\boldsymbol{\Pi}} \quad \sim \quad \mathsf{W}_{\mathbf{a}}.$$

En substituant cette égalité dans l'équation B.5, nous obtenons la *contrainte de consistante* $\mathsf{F} - \mathsf{C}$:

$$\exists \mathbf{a} \in \mathbb{R}^3 \quad | \quad \mathsf{F} \quad \sim \quad [\mathbf{e}']_\wedge ([\mathbf{b}']_\wedge \mathsf{C} + \mathbf{b}'\mathbf{a}^\mathsf{T}).$$

Cette contrainte peut être écrite de manière réciproque, en inversant les rôles de F et de C, pour obtenir la *contrainte de consistance* $\mathsf{C} - \mathsf{F}$. Elle montre que si C est connue, alors la géométrie statique n'a plus que 5 degrés de liberté : la position du plan de mouvement et la position d'un épipôle. Cette équation est non-linéaire pour les inconnues $\mathbf{a}$ et $\mathbf{e}'$. L'estimation linéaire a partir de 5 correspondances de points n'est donc pas possible. Une solution alternative est de considérer que, si le point d'incidence est statique, ses projections $\boldsymbol{b}$ et $\boldsymbol{b}'$ donne une contrainte linéaire sur F, et permettent d'estimer cette dernière avec 6 autres correspondances de points statiques.

B.3.1.2 Calculer $\mathsf{H}_{\boldsymbol{\Pi}}$

Etant donnés la matrice fondamentale et le tenseur-C7, on peut retrouver l'homographie induite par le plan de mouvement. Nous donnons deux méthodes pour cela.

Une solution analytique. Soit $\boldsymbol{X} \in \boldsymbol{\Pi}$ un point dynamique sur le plan de mouvement, et $\boldsymbol{x}$ et $\tilde{\boldsymbol{x}}'$ ses reprojections dans les deux images, $\boldsymbol{x}$ étant connu et $\tilde{\boldsymbol{x}}'$ inconnu. Le point $\tilde{\boldsymbol{x}}'$ se trouve à l'intersection de la droite de mouvement associée au point $\boldsymbol{X}$ dans cette image, et de la droite épipolaire associée au point $\boldsymbol{x}$, ce qui peut être écrit :

$$\tilde{\mathbf{x}}' \quad \sim \quad (\mathsf{C}\mathbf{x}) \wedge (\mathsf{F}\mathbf{x}). \tag{B.6}$$

En général, $\tilde{\mathbf{x}}'$ est donc une fonction quadratique de $\mathbf{x}$. Nous montrons que cette fonction quadratique se réduit à une fonction linéaire, correspondant à l'homographie de plan recherchée $\mathsf{H}_{\boldsymbol{\Pi}}$, si la contrainte de consistance $\mathsf{F} - \mathsf{C}$ est satisfaite. Cette contrainte correspond au fait que F et C sont compatibles avec l'homographie recherchée.

Sous cette hypothèse, elle s'écrivent $\mathsf{C} \sim \mathsf{H}^{\star}_{\boldsymbol{\Pi}}[\mathbf{b}]_{\wedge}$ et $\mathsf{F} \sim \mathsf{H}^{\star}_{\boldsymbol{\Pi}}[\mathbf{e}]_{\wedge}$. Substituons ces écritures dans l'équation (B.6) :

$$\tilde{\mathbf{x}}' \sim [\mathsf{H}^{\star}_{\boldsymbol{\Pi}}[\mathbf{b}]_{\wedge}\mathbf{x}]_{\wedge}\mathsf{H}^{\star}_{\boldsymbol{\Pi}}[\mathbf{e}]_{\wedge}\mathbf{x}.$$

D'après l'équation (2.3) :

$$\tilde{\mathbf{x}}' \sim \mathsf{H}_{\boldsymbol{\Pi}}[[\mathbf{b}]_{\wedge}\mathbf{x}]_{\wedge}\mathsf{H}^{-\mathsf{T}}_{\boldsymbol{\Pi}}\mathsf{H}^{\star}_{\boldsymbol{\Pi}}[\mathbf{e}]_{\wedge}\mathbf{x},$$

et d'après l'équation (2.6) :

$$\tilde{\mathbf{x}}' \sim \mathsf{H}_{\boldsymbol{\Pi}}[[\mathbf{b}]_{\wedge}\mathbf{x}]_{\wedge}[\mathbf{e}]_{\wedge}\mathbf{x}.$$

En supposant que le point $\boldsymbol{x}$ n'est pas sur la droite formée par les points $\boldsymbol{e}$ et $\boldsymbol{b}$, cette équation se réduit à :

$$\tilde{\mathbf{x}}' \sim \mathsf{H}_{\boldsymbol{\Pi}}\mathbf{x}.$$

En effet, elle correspond à relier $\boldsymbol{x}$ à $\boldsymbol{e}$ et à $\boldsymbol{b}$, puis a prendre l'intersection de ces deux droites, qui n'est autre que $\boldsymbol{x}$, ou qui est indéterminée si $\boldsymbol{x}$ est sur la droite formée par $\boldsymbol{e}$ et $\boldsymbol{b}$. En conséquence, nous pouvons écrire :

$$\forall \boldsymbol{x} \in \mathbb{P}^2, \quad (\mathbf{x}^{\mathsf{T}}(\mathbf{e} \wedge \mathbf{b}) \neq 0) \quad \Rightarrow \quad (\mathsf{H}_{\boldsymbol{\Pi}}\mathbf{x} \sim (\mathsf{C}\mathbf{x}) \wedge (\mathsf{F}\mathbf{x})).$$

En appliquant cette équation pour les 4 vecteurs de la base canonique de $\mathbb{P}^2$, nous obtenons la formule suivante :

$$\mathsf{H}_{\boldsymbol{\Pi}} \sim (\mathsf{C} \wedge \mathsf{F})\,\mathrm{diag}\left((\mathsf{C} \wedge \mathsf{F})^{-1}[\sum_k \mathbf{c}_k]_{\wedge}\left(\sum_k \mathbf{f}_k\right)\right),$$

où les $\mathbf{c}_k$ et les $\mathbf{f}_k$ sont les colonnes de C et de F respectivement, et où $\mathsf{C} \wedge \mathsf{F}$ est la matrice formée par le produit vectoriel des colonnes de C et F.

Cette équation doit être utilisée avec prudence. En effet, si C et F sont estimés de manière consistante, $\mathsf{H}_{\boldsymbol{\Pi}}$ est parfaitement définie. Par contre, si ils sont estimés indépendamment, la solution analytique donne une solution approchée basée sur l'erreur de transfert induite par les 4 points de la base canonique de $\mathbb{P}^2$. Parmi ces 4 points, 2 sont à l'infini, et ne sont donc pas appropriés à l'estimation d'une homographie destinée à être utilisée sur des points image. De plus, si l'un de ces 4 points est sur la droite entre $\boldsymbol{e}$ et $\boldsymbol{b}$, l'équation dégénère.

Une solution géométrique. L'idée est de trouver un moyen pour utiliser les algorithmes standards d'estimation d'une homographie, présentés en §2.6.1, chapitre 2. Nous proposons de générer des correspondances de points statiques, à partir des points dynamiques, à l'aide de l'équation (B.6), c'est à dire par intersection des droites de mouvement et des droites épipolaires. L'algorithme d'estimation peut être résumé comme suit :

1. génération de correspondances de points statiques sur le plan de mouvement :

$$\{\cdots,\ \mathbf{x} \leftrightarrow ((\mathsf{C}\mathbf{x}) \wedge (\mathsf{F}\mathbf{x})), ((\mathsf{C}^{\mathsf{T}}\mathbf{x}') \wedge (\mathsf{F}^{\mathsf{T}}\mathbf{x}')) \leftrightarrow \mathbf{x}', \cdots\}\ ;$$

2. estimation de $\mathsf{H}_{\boldsymbol{\Pi}}$ à partir de ces correspondances, par une méthode standard.

Nous avons choisi de minimiser l'erreur de transfert Euclidienne, à l'aide de l'algorithme de Levenberg-Marquardt, initialisé avec la solution analytique précédemment décrite. La figure B.6 montre le résultat obtenu en estimant ces homographies entre paires d'images, et en transférant la position des points dynamiques d'une image à l'autre.

B.3.2 Le cas multi-vues

La section précédente montre que deux vues fixent complètement la géométrie dynamique car le point d'incidence, de même que le plan de mouvement, peuvent être reconstruits de manière unique. Concernant la structure dynamique, chaque nouvelle vue ajoute 1 degré de liberté supplémentaire à chaque point, correspondant à sa position le long de sa droite de mouvement, à l'instant de la prise de vue, comme dans le cas purement dynamique.

Nous avons vu au chapitre 2, §2.10, que la géométrie d'un ensemble de n caméras pouvait être représentée par une matrice fondamentale et $n-2$ matrices de projection. Nous examinons ci-dessous le lien entre le tenseur-C et une matrice de projection. Nous montrons comment contraindre l'estimation d'une matrice de projection par la connaissance d'un tenseur-C.

FIG. B.6 – Les points dynamiques sont transférés entre les images en utilisant les homographies induites par le plan de mouvement, c'est à dire comme si ils étaient statiques. Ces homographies sont calculées par la méthode de génération de correspondances statiques (voir texte) et la minimisation non-linéaire de la distance Euclidienne symétrique entre des points transférés et mesurés. L'erreur finale est 0,35 pixels. La solution initiale donnée par la solution analytique pour l'homographie (non montrée ici) est 2,77 pixels.

B.3.2.1 Le lien entre C et P

Considérons un ensemble d'images, pour lesquelles les géométries dynamiques et statiques sont connues. Sans perte de généralité, on suppose que $\mathsf{P} \sim (\ \mathrm{I}\quad \mathbf{0})$ et $\mathsf{P}' \sim (\ \mathsf{H}_{\boldsymbol{\Pi}}\quad \mathbf{e}')$, où $\mathsf{H}_{\boldsymbol{\Pi}}$ est l'homographie de plan induite par le plan de mouvement entre la vue 1 et la vue 2. Nous désirons exprimer le lien entre une troisième vue, dont la matrice de projection est P'', et un tenseur-C5 ou un tenseur-C7 C décrivant la géométrie dynamique entre la vue 2 et la vue 3. Avec le choix précédent pour P et P', on peut écrire :

$$\mathsf{P}'' \quad \sim \quad (\ \mathsf{H}'_{\boldsymbol{\Pi}}\quad \mathbf{e}'),$$

où $\mathsf{H}'_{\boldsymbol{\Pi}}$ est l'homographie induite par le plan de mouvement entre la vue 1 et la vue 3, et $\boldsymbol{e}'$ est l'épipôle de la vue 1 dans la vue 3. L'homographie $\mathsf{H}'_{\boldsymbol{\Pi}}$ peut s'écrire en composant l'homographie $\mathsf{H}_{\boldsymbol{\Pi}}$ (connue) entre la vue 1 et la vue 2 et l'homographie entre la vue 2 et la vue 3, notée $\mathsf{H}''_{\boldsymbol{\Pi}}$. Cette dernière est inconnue. En utilisant le même raisonnement qu'en §B.3.1.1, on l'exprime en terme du tenseur-C entre ces deux vues (connu) :

$$\exists \mathbf{a} \in \mathbb{R}^3 \quad | \quad \mathsf{H}''_{\boldsymbol{\Pi}} \quad \sim \quad \mathsf{W}_{\mathbf{a}},$$

où $\mathsf{W}_{\mathbf{a}}$ est la paramétrisation (B.2) des homographies 2D compatibles avec le tenseur-C :

$$\mathsf{W}_{\mathbf{a}} \quad \sim \quad [\mathbf{b}']_{\wedge}\mathsf{C} + \mathbf{b}'\mathbf{a}^{\mathsf{T}}.$$

On obtient donc l'expression suivante pour $\mathsf{H}'_{\boldsymbol{\Pi}}$:

$$\exists \mathbf{a} \in \mathbb{R}^3 \quad | \quad \mathsf{H}'_{\boldsymbol{\Pi}} \quad \sim \quad \mathsf{H}_{\boldsymbol{\Pi}} \underbrace{([\mathbf{b}']_{\wedge}\mathsf{C} + \mathbf{b}'\mathbf{a}^{\mathsf{T}})}_{\mathsf{W}_{\mathbf{a}}},$$

et la *contrainte de consistance* $\mathsf{P} - \mathsf{C}$:

$$\exists \mathbf{a} \in \mathbb{R}^3 \quad | \quad \mathsf{P}'' \quad \sim \quad (\ \mathsf{H}_{\boldsymbol{\Pi}}([\mathbf{b}']_{\wedge}\mathsf{C} + \mathbf{b}'\mathbf{a}^{\mathsf{T}}) \quad \mathbf{e}''). \tag{B.7}$$

B.3.2.2 Estimation

Deux cas d'estimation peuvent être considérés. Le premier est celui de l'estimation d'un tenseur-C étant donnée la matrice de projection, et le deuxième cas est le contraire. Nous considérons le deuxième cas : étant donné le tenseur-C, encapsulant la géométrie dynamique de la vue, calculer la matrice de projection P'' de cette vue, à partir de correspondances de points statiques 3D/2D notées $\boldsymbol{Q} \leftrightarrow \boldsymbol{q}$. L'approche standard est de considérer l'équation de projection $\mathbf{q} \sim \mathsf{P}''\mathbf{Q}$. Chaque correspondance fournit 2 équations, et les 11 degrés de libertés de P'' sont déterminés avec 6 correspondances de point. Nous utilisons la même idée, mais paramétrons

P'' à l'aide de la contrainte de consistance $P - C$. Les inconnues sont les vecteurs $\mathbf{a}$ et $\mathbf{e}'$ pour lesquelles la paramétrisation de P'' est linéaire. Nous pouvons donc construire un système linéaire à partir de l'équation de projection. Ce système à une solution unique à partir de 3 correspondances de points. On peut ensuite minimiser l'erreur de reprojection en utilisant une méthode d'optimisation non-linéaire.

B.4 Résultats expérimentaux

Nous considérons la séquence de la route, dont des images sont montrées en figure B.1. Nous sélectionnons manuellement des points statiques et dynamiques dans la première image, et lançons une procédure de suivi automatique au travers de la séquence, basée sur des mesures de corrélation. Une technique d'estimation robuste basée sur RANSAC [64] permet d'éliminer les fausses correspondances dans les procédures d'estimation utilisées ci-dessous.

Nous commençons par estimer la géométrie dynamique de la scène, en lançant un calcul séquentiel d'un réseau de tenseurs-C, voir §B.2 : un tenseur-C7 est estimé entre les deux premières images, puis une succession de tenseurs-C5 est calculée entre les images consécutives suivantes. Ensuite, nous estimons la géométrie statique de la scène, comme indiqué en §B.3 : nous calculons une matrice fondamentale entre les deux premières images, contrainte par le tenseur-C7, puis reconstruisons les points statiques par triangulation. Nous calculons ensuite une matrice de projection pour chacune des images suivantes, contrainte par le tenseur-C5 correspondant. Finalement, nous retrouvons les homographies inter-images induites par le plan de mouvement, comme indiqué en §B.3.1.2. Ces homographies permettent de transférer les points dynamiques, et de prédire leur position d'une image à une autre, comme si ils étaient statiques. Le résultat de tels transferts est montré sur la figure B.7. Cette figure montre que l'homographie estimée entre les images 0 et 10 est visuellement précise, car le transfert des parties statiques est visuellement bon.

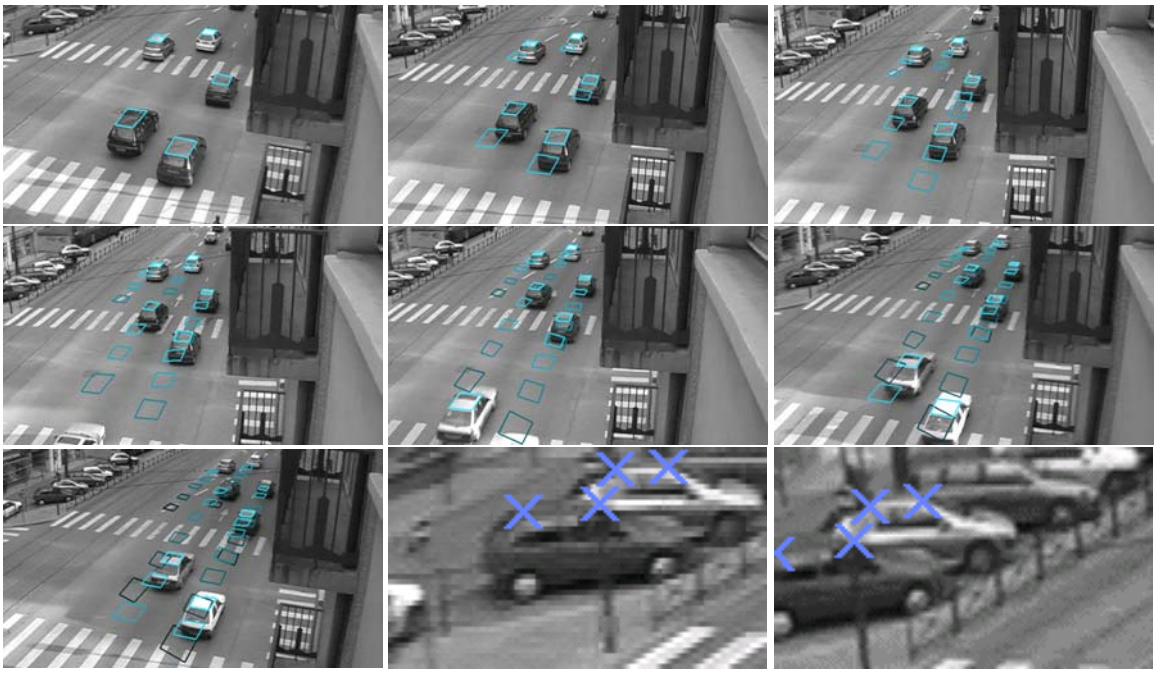


FIG. B.7 – Les homographies inter-images induites par le plan de mouvement sont retrouvées à partir de correspondances de points dynamiques seulement. Elle permettent de transférer la position des véhicules en mouvement d'une image à une autre, comme si ces derniers étaient statiques. Les deux dernières images sont des zooms sur les images 0 et 10 respectivement. La première montre des points statiques sur le plan de mouvement (sélectionnés manuellement), et la deuxième montre le transfert de ces points à l'aide de l'homographie induite par le plan de mouvement, calculée à l'aide de points dynamiques seulement.

Le principal problème rencontré pour cette expérimentation est le calcul du tenseur-C7 initial, entre les deux premières images. En effet, nous pouvons observer que tous les véhicules roulent globalement à la même allure, ce qui induit une homographie mettant en correspondance les points dynamiques entre des deux vues. Il y a donc une ambiguïté de dimension 2 pour le calcul du tenseur-C7. Pour résoudre cette ambiguïté, nous avons calculé un tenseur-C5. La position du point d'incidence dans la première image est obtenue en intersectant les droites induites par les bandes blanches dessinées sur la route.

B.5 Conclusions et perspectives

Conclusions. Nous avons examiné un scénario dynamique décrivant le mouvement de points le long de droites de mouvement, convergants vers un point d'incidence et appartenant à un plan de mouvement. Nous montrons qu'il existe un tenseur d'appariement simple, le tenseur-C, similaire à la matrice fondamentale. Dans le cas de deux vues, ce tenseur encapsule la géométrie dynamique, c'est à dire les projections du point d'incidence dans les deux images, et une homographie 1D entre les droites de mouvement. Ce tenseur est désigné plus précisémment par le terme *tenseur-C7*. Nous montrons que la géométrie dynamique de n vues est modélisée par un tenseur-C7 et $n-2$ tenseurs-C5. Un *tenseur-C5* est une spécialisation du tenseur-C7, qui n'encapsule qu'une seule projection du point d'incidence. Nous donnons des algorithmes d'estimation consistante de cette géométrie. Nous montrons comment la géométrie dynamique est liée à la géométrie statique standard des caméras, et donnons des algorithmes permettant d'estimer la géométrie statique de manière contrainte par la géométrie dynamique. Nous obtenons un résultat consistant, et nécessitant moins de correspondances de points que les algorithmes standards. Nous montrons comment reconstruire le plan de mouvement, à partir de correspondances de points dynamiques seulement. Les résultats expérimentaux montrent que notre approche s'applique à l'estimation de la géométrie induite par des scènes réelles, et peut être utilisée en pratique pour la modélisation de caméras vidéos de surveillance observant des routes.

Perspectives. Les méthodes proposées nécessitent la segmentation des points mesurés en points dynamiques et en points statiques. On peut imaginer rendre cette segmentation automatique, par exemple avec l'estimation robuste successive de la matrice fondamentale puis du tenseur-C, en utilisant les correspondances rejetées par la première estimation, comme données d'entrée pour la deuxième. Un des problèmes avec cette approche est que les structures du tenseur-C et de la matrice fondamentale étant très proches, on ne peut distinguer lequel des deux est estimé en premier, dans le cas général. Une idée de solution par rapport à cette ambiguïté est de considérer des triplets de vues, car dans ce cas, nous avons vu que la modélisation de la géométrie dynamique était différente de celle de la géométrie statique.

Une autre approche pour la segmentation est proposée par Wolf et Shashua [229]. L'idée est de trouver un tenseur valide pour les points dynamiques et statiques. Considérons une correspondance de point $\boldsymbol{u} \leftrightarrow \boldsymbol{u}'$, issue d'un point statique ou dynamique. Si ce point est statique, alors la correspondance doit satisfaire l'équation fondamentale $\mathbf{u}'^{\mathsf{T}}\mathsf{F}\mathbf{u} = 0$, et si il est dynamique, alors elle doit satisfaire l'équation (B.1), $\mathbf{u}'^{\mathsf{T}}\mathsf{C}\mathbf{u} = 0$. En conséquence, la contrainte suivante doit être vérifée :

$$\left(\mathbf{u}'^{\mathsf{T}}\mathsf{F}\mathbf{u}\right)\left(\mathbf{u}'^{\mathsf{T}}\mathsf{C}\mathbf{u}\right) \;=\; 0.$$

Wolf et Shashua montrent que si l'on développe cette équation, et après quelques manipulations algébriques, on obtient :

$$\mathcal{U}'\mathcal{S}\mathcal{U} \;=\; 0,$$

où $\mathcal{S}$ est une matrice (6×6) appelée par Wolf et Shashua la "matrice de segmentation". Le vecteur $\mathcal{U}$ représente les coordonnées de $\boldsymbol{u}$, "relevées" sur $\mathbb{P}^5$, données par :

$$\mathcal{U}^{\mathsf{T}} \;\sim\; (u_1^2 \;\; u_1u_2 \;\; u_2^2 \;\; u_1u_3 \;\; u_2u_3 \;\; u_3^2).$$

Ce tenseur a la propriété d'encapsuler intégralement les géométries dynamiques et statiques de notre scénario. Cependant, il n'est pas bien adapté à l'estimation robuste, car son estimation linéaire nécessite 35 correspondances de points, parmi lesquelles un minimum de 8 points statiques et de 8 points dynamiques sont nécessaires,

ce qui augmente le nombre de tirages d'un algorithme robuste par échantillonage aléatoire tel RANSAC [64]. L'extraction du tenseur-C et de la matrice fondamentale à partir de $\mathcal{S}$ est une opération difficile. Notons que Vidal et al. [223, 224] généralisent cette approche.

Finalement, on note qu'en pratique, les droites de mouvement sont souvent parallèles. Le point d'incidence est donc à l'infini, ce qui peut être exploité lors d'une éventuelle étape de calibrage en ligne des caméras.

ANNEXE

C

CORRECTION DE PLÜCKER

C.1 Introduction

Cette annexe est liée au chapitre 6 concernant la reconstruction de droites et de caméras. Plus particulièrement, elle aborde le problème de la *correction de Plücker*, nécessaire pour les algorithmes de triangulation de droites présentés en §6.3, chapitre 6. Ces algorithmes utilisent les coordonnées de Plücker pour représenter les droites. Pour des raisons de commodité, la contrainte de Plücker est parfois négligée. On peut faire l'analogie avec l'algorithme des huit points pour le calcul de la matrice fondamentale : la contrainte de rang deux est négligée dans un premier temps, puis la matrice estimée est corrigée. Dans les deux sections suivantes, nous suggérons un critère de correction et formulons algébriquement le problème, puis donnons un algorithme de résolution.

C.2 Formulation du problème

Soit $\mathbf{L}^\mathsf{T} \sim (\ \mathbf{a}^\mathsf{T} \quad \mathbf{b}^\mathsf{T}\ )$ un vecteur-6 ne satisfaisant pas nécessairement la contrainte de Plücker (2.55), c'est à dire que $\mathbf{a}^\mathsf{T}\mathbf{b}$ peut ne pas être nul. Nous cherchons un vecteur $\hat{\mathbf{L}}^\mathsf{T} \sim (\ \mathbf{u}^\mathsf{T} \quad \mathbf{v}^\mathsf{T}\ )$ tel que $\mathbf{u}^\mathsf{T}\mathbf{v} = 0$ et tel qu'un certain critère de correction soit minimisé. Pour choisir ce critère de correction, utilisons l'analogie avec l'algorithme des huit points. Soit F la matrice estimée de manière linéaire. La correction du rang est effectuée en trouvant la matrice de rang deux $\hat{\mathsf{F}}$ telle que la distance entre F et $\hat{\mathsf{F}}$ soit minimisée. Cette distance s'exprime à l'aide de la norme de Frobenius, et le problème de la correction de rang se formule comme :

$$\min_{\hat{\mathsf{F}},\det\hat{\mathsf{F}}=0} \|\mathsf{F}-\hat{\mathsf{F}}\|^2.$$

Une solution élégante, basée sur la décomposition de F en valeurs singulières, est possible, voir [86, 233].

Dans le cas des coordonnées de Plücker, nous formulons le problème similairement par :

$$\min_{\hat{\mathbf{L}},\mathbf{u}^\mathsf{T}\mathbf{v}=0} \mathcal{O} \quad \text{avec} \quad \mathcal{O} \;=\; \|\mathbf{L}-\hat{\mathbf{L}}\|^2. \tag{C.1}$$

Ce problème a une formulation simple et concise, mais il n'est pas trivial.

Notons que Kanatani [110] puis Leedan et Meer [120] proposent une méthode de correction itérative générale. L'idée est de linéariser la contrainte requise, puis de projeter le vecteur sur sous-espace ainsi défini. Ces deux étapes sont répétées jusqu'à convergence.

C.3 Résolution

Nous décrivons notre solution au problème de la correction de Plücker tel qu'il est posé dans la section précédente. L'algorithme correspondant est donné dans le tableau C.1. Nous transformons le problème original en un problème réduit équivalent.

Une interprétation géométrique. Interprétons les vecteurs $\mathbf{a}$ et $\mathbf{b}$ comme les coordonnées de deux points 3D $\boldsymbol{a}$ et $\boldsymbol{b}$. Ces deux points ne sont pas directement liés avec la droite 3D $\boldsymbol{L}$ sous-jacente. Cette interprétation est simplement destinée à mieux visualiser le problème. De même, interprétons les vecteurs $\mathbf{u}$ et $\mathbf{v}$ formant les coordonnées de Plücker corrigées, comme les coordonnées de deux points 3D $\boldsymbol{u}$ et $\boldsymbol{v}$. La contrainte de Plücker $\mathbf{u}^\mathsf{T}\mathbf{v} = 0$ s'interprète comme l'orthogonalité des droites passant par l'origine d'une part, et par les points $\boldsymbol{u}$ et $\boldsymbol{v}$ respectivement. Le critère de correction (C.1) donne la somme des carrés des distances Euclidiennes entre $\boldsymbol{a}$ et $\boldsymbol{u}$ et entre $\boldsymbol{b}$ et $\boldsymbol{v}$. Le problème est donc de trouver deux points $\boldsymbol{u}$ et $\boldsymbol{v}$, les plus proches possibles de $\boldsymbol{a}$ et $\boldsymbol{b}$ respectivement, sous la contrainte que les deux droites formées par l'origine et $\boldsymbol{u}$ et $\boldsymbol{v}$ respectivement, soient orthogonales. Nous utilisons la stratégie de résolution suivante. La première étape consiste à aligner le plan formé par l'origine, $\boldsymbol{a}$ et $\boldsymbol{b}$, avec le plan $z = 0$. C'est la *réduction du problème* : on passe d'un problème 3D à un problème 2D. La deuxième étape consiste à trouver deux points dans le plan $z = 0$ minimisant le critère de correction et engendrant des droites orthogonales avec l'origine. C'est la *résolution du problème réduit*. Finalement, on exprime la solution obtenue dans l'espace 3D original, c'est l'*expression de la solution*.

Réduction du problème. Définissons les matrices (3×2) $\bar{\mathsf{C}} \sim (\mathbf{a}\ \ \mathbf{b})$ et $\hat{\mathsf{C}} \sim (\mathbf{u}\ \ \mathbf{v})$ encapsulant respectivement les vecteurs $\mathbf{L}$ et $\hat{\mathbf{L}}$. La contrainte de Plücker (2.55) est vérifiée si et seulement si la matrice $\hat{\mathsf{C}}$ a des colonnes orthogonales : la matrice $\hat{\mathsf{C}}^\mathsf{T}\hat{\mathsf{C}}$ doit être diagonale. Reécrivons le critère de correction en termes des matrices $\bar{\mathsf{C}}$ et $\hat{\mathsf{C}}$:

$$\mathcal{O} \;=\; \|\bar{\mathsf{C}}-\hat{\mathsf{C}}\|^2.$$

Calculons la décomposition en valeurs singulières de $\bar{\mathsf{C}}_{(3\times2)}$:

$$\bar{\mathsf{C}}_{(3\times2)} \;=\; \underbrace{\begin{pmatrix} \bar{\mathsf{U}}_{(3\times2)} & \bar{\mathbf{u}}_{(3\times1)} \end{pmatrix}}_{\mathsf{U}_{(3\times3)}} \underbrace{\begin{pmatrix} \bar{\mathsf{\Sigma}}_{(2\times2)} \\ 0 \quad 0 \end{pmatrix}}_{\mathsf{\Sigma}_{(3\times2)}} \bar{\mathsf{V}}^\mathsf{T}_{(2\times2)}. \tag{C.2}$$

En suivant l'interprétation géométrique du paragraphe précédent, la matrice U donne la rotation 3D[1] permettant d'amener les points $\boldsymbol{a}$ et $\boldsymbol{b}$ sur le plan $z = 0$. Cette décomposition permet d'écrire :

$$\mathcal{O} \;=\; \|\mathsf{U}\bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}} - \hat{\mathsf{C}}\|^2 \;=\; \|\bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}} - \mathsf{U}^{\mathsf{T}}\hat{\mathsf{C}}\|^2,$$

car $\mathsf{U} \in O(3)$ préserve la norme $\mathcal{L}_2$. En considérant plus spécifiquement la forme de la décomposition (C.2), on obtient :

$$\begin{aligned}\mathcal{O} \;&=\; \left\| \begin{pmatrix} \bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}} \\ 0 \quad 0 \end{pmatrix} - \underbrace{\begin{pmatrix} \bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}} \\ \bar{\mathbf{u}}^{\mathsf{T}}\hat{\mathsf{C}} \end{pmatrix}}_{\mathsf{Y}_{(3\times 2)}} \right\|^2 \\ &=\; \|\bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}} - \bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}}\|^2 + \|\bar{\mathbf{u}}^{\mathsf{T}}\hat{\mathsf{C}}\|^2.\end{aligned}$$

Les deux zéros correspondent au fait que les points $\boldsymbol{a}$ et $\boldsymbol{b}$ transformés par la rotation U se trouvent sur le plan $z = 0$. Comme précédemment annoncé, la solution recherchée se trouve de même dans le plan $z = 0$, d'où $\bar{\mathbf{u}}^{\mathsf{T}}\hat{\mathsf{C}} = \mathbf{0}^{\mathsf{T}}_{(2\times 1)}$, et le critère de correction réduit suivant :

$$\mathcal{O} \;=\; \|\bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}} - \bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}}\|^2.$$

Notons $\bar{\mathsf{Z}} = \bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}}$ et $\hat{\mathsf{Z}} = \bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}}$. La matrice $\bar{\mathsf{V}}$ est orthonormale et $\bar{\Sigma}$ est diagonale. La matrice $\bar{\mathsf{Z}}$ a donc des lignes orthogonales ($\bar{\mathsf{Z}}\bar{\mathsf{Z}}^{\mathsf{T}}$ est diagonale, mais $\bar{\mathsf{Z}}^{\mathsf{T}}\bar{\mathsf{Z}}$ ne l'est pas pas). Le problème est donc réduit à l'ajustement d'une matrice $\hat{\mathsf{Z}}$ avec des colonnes orthogonales à une matrice $\bar{\mathsf{Z}}$ avec des lignes orthogonales[2]. Notons que $\hat{\mathsf{Z}} = \bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}}$ implique la relation $\hat{\mathsf{C}} = \bar{\mathsf{U}}\hat{\mathsf{Z}}$ en dépit du fait que $\bar{\mathsf{U}}\bar{\mathsf{U}}^{\mathsf{T}} \neq \mathrm{I}_{(3\times 3)}$[3].

Résolution du problème réduit. On paramétrise la matrice $\hat{\mathsf{Z}}$ dont les colonnes sont orthogonales par $\hat{\mathsf{V}}\hat{\Sigma}$, où $\hat{\mathsf{V}}$ est orthonormale et $\hat{\Sigma}$ est diagonale. On peut donc écrire :

$$\mathcal{O} \;=\; \|\bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}} - \hat{\mathsf{V}}\hat{\Sigma}\|^2 \;=\; \|\hat{\mathsf{V}}^{\mathsf{T}}\bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}} - \hat{\Sigma}\|^2.$$

La matrice diagonale $\hat{\Sigma}$ qui minimise cette expression est donnée par les entrées diagonales de la matrice $\hat{\mathsf{V}}^{\mathsf{T}}\bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}}$, quelle que soit la solution pour $\hat{\mathsf{V}}$. Le problème est maintenant de trouver la matrice orthonormale $\hat{\mathsf{V}} = (\hat{\mathbf{v}}_1 \;\; \hat{\mathbf{v}}_2)$ telle que la somme des carrés des entrées non-diagonales du produit $\hat{\mathsf{V}}^{\mathsf{T}}\bar{\mathsf{Z}}$, avec $\bar{\mathsf{Z}} = \bar{\Sigma}\bar{\mathsf{V}}^{\mathsf{T}} = (\mathbf{z}_1 \;\; \mathbf{z}_2)$, soit minimisée :

$$\mathcal{O} \;=\; (\hat{\mathbf{v}}_1^{\mathsf{T}}\mathbf{z}_2)^2 + (\hat{\mathbf{v}}_2^{\mathsf{T}}\mathbf{z}_1)^2.$$

Définissons la matrice de rotation 2D d'angle $\pi/2$ $\mathsf{M} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$. On paramétrise la matrice orthonormale $\hat{\mathsf{V}}$ par un vecteur unitaire $\hat{\mathbf{v}}$:

$$\begin{cases} \hat{\mathbf{v}}_1 \;=\; \hat{\mathbf{v}} \\ \hat{\mathbf{v}}_2 \;=\; \mathsf{M}\hat{\mathbf{v}}, \end{cases}$$

Le critère de correction se reécrit donc comme :

$$\mathcal{O} \;=\; (\hat{\mathbf{v}}^{\mathsf{T}}\mathbf{z}_2)^2 + (\hat{\mathbf{v}}^{\mathsf{T}}\mathsf{M}^{\mathsf{T}}\mathbf{z}_1)^2 \;=\; \|\mathsf{T}\hat{\mathbf{v}}\|^2 \quad \text{avec} \quad \mathsf{T} \;=\; \begin{pmatrix} \mathbf{z}_2^{\mathsf{T}} \\ \mathbf{z}_1^{\mathsf{T}}\mathsf{M} \end{pmatrix}.$$

Le vecteur unitaire $\hat{\mathbf{v}}$ qui minimise cette expression est donné par le vecteur singulier associé à la plus petite valeur singulière de T.

[1] En fait, U est une matrice de $O(3)$ dont le déterminant est ± 1. Pour garantir $\mathsf{U} \in SO(3)$, on peut multiplier U par $\det(\mathsf{U})$. Toutes fois, l'algorithme est valable même si $\mathsf{U} \notin SO(3)$.

[2] Le fait que la matrice $\hat{\mathsf{Z}} = \bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}}$ ait des colonnes orthogonales provient de la contrainte de Plücker. En effet, cette contrainte implique que $\hat{\mathsf{C}}$ a des colonnes orthogonales, donc $\hat{\mathsf{C}}^{\mathsf{T}}\hat{\mathsf{C}}$ est diagonale. La matrice $\mathsf{U}^{\mathsf{T}}\hat{\mathsf{C}}$ a aussi des colonnes orthogonales. Or, $\hat{\mathsf{C}}^{\mathsf{T}}\mathsf{U}\mathsf{U}^{\mathsf{T}}\hat{\mathsf{C}} = \hat{\mathsf{C}}^{\mathsf{T}}\bar{\mathsf{U}}\bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}} + \hat{\mathsf{C}}^{\mathsf{T}}\bar{\mathbf{u}}\bar{\mathbf{u}}^{\mathsf{T}}\hat{\mathsf{C}} = \hat{\mathsf{C}}^{\mathsf{T}}\bar{\mathsf{U}}\bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}}$ car $\bar{\mathbf{u}}^{\mathsf{T}}\hat{\mathsf{C}} = \mathbf{0}^{\mathsf{T}}_{(2\times 1)}$, d'où le fait que $\bar{\mathsf{U}}^{\mathsf{T}}\hat{\mathsf{C}}$ ait des colonnes orthogonales.

[3] En effet, $\mathsf{Y} = \mathsf{U}^{\mathsf{T}}\hat{\mathsf{C}}$, d'où $\hat{\mathsf{C}} = \mathsf{U}\mathsf{Y}$. Comme $\mathsf{Y}^{\mathsf{T}} = (\;\hat{\mathsf{Z}}^{\mathsf{T}} \quad \mathbf{0}_{(2\times 1)})$, $\hat{\mathsf{C}} = \mathsf{U}\mathsf{Y} = \bar{\mathsf{U}}\hat{\mathsf{Z}}$.

Expression de la solution. A partir du vecteur $\hat{\mathbf{v}}$ solution du problème réduit, on forme la matrice orthonormale $\hat{\mathsf{V}} = \begin{pmatrix} \hat{v}_1 & -\hat{v}_2 \\ \hat{v}_2 & \hat{v}_1 \end{pmatrix}$. La matrice diagonale $\hat{\Sigma}$ est donnée par $\hat{\Sigma} = \mathrm{diag}(\hat{\mathsf{V}}^\mathsf{T}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T})$. On retrouve ensuite $\hat{\mathsf{Z}} = \hat{\mathsf{V}}\hat{\Sigma} = \hat{\mathsf{V}}\,\mathrm{diag}(\hat{\mathsf{V}}^\mathsf{T}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T})$ puis $\hat{\mathsf{C}} = \bar{\mathsf{U}}\hat{\mathsf{Z}} = \bar{\mathsf{U}}\,\hat{\mathsf{V}}\,\mathrm{diag}(\hat{\mathsf{V}}^\mathsf{T}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T})$. Le tableau C.1 résume notre algorithme de correction de Plücker.

1. calculer la décomposition en valeurs singulières $\bar{\mathsf{C}} \sim (\mathbf{a}\ \ \mathbf{b}) \sim \bar{\mathsf{U}}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T}$;
2. soit $\bar{\mathsf{Z}} = \bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T}$, former la matrice $\mathsf{T} \sim \begin{pmatrix} z_{21} & z_{22} \\ z_{12} & -z_{11} \end{pmatrix}$;
3. calculer le vecteur singulier $\hat{\mathbf{v}}$ associé à la plus petite valeur singulière de T ;
4. soit $\hat{\mathsf{V}} = \begin{pmatrix} \hat{v}_1 & -\hat{v}_2 \\ \hat{v}_2 & \hat{v}_1 \end{pmatrix}$, on obtient : $(\mathbf{u}\ \ \mathbf{v}) \sim \bar{\mathsf{U}}\,\hat{\mathsf{V}}\,\mathrm{diag}\left(\hat{\mathsf{V}}^\mathsf{T}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T}\right)$.

TAB. C.1 – Algorithme de correction de Plücker : étant donné un vecteur-6 $\mathbf{L}^\mathsf{T} \sim (\ \mathbf{a}^\mathsf{T}\quad \mathbf{b}^\mathsf{T}\ )$, cet algorithme calcule les plus proches coordonnées de Plücker $\hat{\mathbf{L}}^\mathsf{T} \sim (\ \mathbf{u}^\mathsf{T}\quad \mathbf{v}^\mathsf{T}\ )$, $\mathbf{u}^\mathsf{T}\mathbf{v} = 0$, au sens de la norme $\mathcal{L}_2$, c'est à dire que $\|\hat{\mathbf{L}} - \mathbf{L}\|$ est minimisé.

ANNEXE

D

AUTRES TRAVAUX

Nous décrivons briévement certains des travaux effectués durant notre période de thèse et absents du corps principal du manuscrit. D'autres travaux, non décrits ci-dessous, menés en collaboration avec Marc-André Ameller et Long Quan, concernant la reconstruction minimale de caméras à partir d'un modèle de projection affine calibré, ont été publiés dans [2, 3]. Une étude d'un système d'ajustement de faisceaux métrique indépendant des paramètres internes des caméras, menée en collaboration avec Ezio Malis, a été publiée dans [131]. Une présentation du projet de recherche VISIRE dans lequel j'ai été impliqué a été publié dans [167]. Finalement, nous avons proposé dans [73], en collaboration avec Nicolas Guilbert, une méthode de reconstruction de caméras basée sur les contraintes de fermetures et exploitant la structure éparse du problème.

Ajustement de faisceaux invariant à la jauge. Nous l'avons mentionné dans les perspectives de la thèse : la plupart des algorithmes d'ajustement de faisceaux ne sont pas invariants à la jauge. La jauge est liée entre autres à la base de l'espace dans laquelle est exprimée le modèle 3D. Une itération d'un algorithme d'ajustement de faisceaux, opérant sur des données représentant la même géométrie, mais exprimées dans deux bases différentes de l'espace 3D, peut donner des résultats très différents. De manière surprenante, la plupart des travaux existants ne considèrent pas ce problème : la jauge est laissée libre, ou une jauge triviale est utilisée. Nous renvoyons au chapitre 3, §3.4 pour plus de détails. L'idée exploitée pour ce travail est la suivante. Si différentes jauges donnent différents résultats, alors il existe une (ou plusieurs) jauges donnant un résultat meilleur que les autres, c'est à dire produisant une minimisation plus importante de l'erreur de reprojection au cours de l'itération. Notre contribution est une méthode permettant de trouver une telle jauge. Pour ce faire, nous formulons une expression de l'erreur de reprojection après l'itération, en fonction de la transformation de jauge. La minimisation de l'approximation au premier ordre de cette erreur nous donne la transformation de jauge recherchée. Notons que cette transformation de jauge conduit à la meilleure jauge (au premier ordre) de manière invariante à la jauge actuelle dans laquelle est exprimées le modèle 3D. Notre algorithme est donc invariant à la jauge et maximise la minimisation de l'erreur de reprojection à chaque itération (au premier ordre). Des résultats expérimentaux préliminaires montrent que notre algorithme converge plus rapidement que les algorithmes standards d'ajustement de faisceaux. Ces travaux ont été publiés dans [15].

Des séquences vidéo aux panoramas de mouvement. Nous décrivons une méthode de construction automatique de mosaïques ou panoramas à partir de séquences vidéo. Les travaux précédents sur les mosaïques d'images se concentrent essentiellement sur le cas de scènes statiques, observées par une caméra en mouvement de rotation pure. Nous généralisons ces approches aux cas où les scènes observées sont constituées d'une partie statique et d'une partie dynamique. Nous décrivons une technique robuste pour l'alignement précis d'un grand nombre d'images, lorsque le mouvement de la caméra et ses paramètres internes sont inconnus. Cette technique d'alignement combine une méthode basée sur des correspondances de points et une méthode basée sur la couleur de tous les pixels. La méthode basée sur les correspondances de points permet, après une estimation initiale et un ajustement de faisceaux sur les paramètres du mouvement, la segmentation des images en parties statiques et dynamiques. Cette étape est suivie d'un alignement précis basé sur la couleur des pixels. Un panorama de mouvement est alors construit. Il est constitué d'un panorama statique, représentant les parties statiques des images, et de la partie dynamique de chaque image. Nous validons notre approche sur de longues séquences d'images d'évenements athlétiques. Ces travaux ont été effectués en collaboration avec Navneet Dalal, Biswajit Bose et Radu Horaud, et publiés dans [16, 17, 18].

Une méthode d'ajustement de faisceaux quasi-linéaire. Nous présentons un algorithme d'ajustement de faisceaux, minimisant l'erreur de reprojection, et non basé sur une technique d'optimisation non-linéaire dérivée de la méthode de Newton, comme les algorithmes standards. Cet algorithme est basé sur le découplage de chaque itération en deux phases : le calcul du mouvement ou *resection* puis le calcul de la structure ou *intersection*. L'originalité de notre approche est de résoudre chacune de ces phases à l'aide d'une méthode de résolution quasi-linéaire, voir le chapitre 2, §2.6.2, permettant de minimiser l'erreur de reprojection. Nous utilisons une approche stratifiée, et formulons l'algorithme pour des caméras perspectives et affines, calibrées et non calibrées. Les algorithmes obtenus s'implantent très simplement. Les résultats expérimentaux montrent qu'ils donnent des résultats comparables aux algorithmes standards en termes de précision, et sont plus rapides, tant que le nombre d'images reste inférieur à la centaine. Ces travaux ont été publiés dans [14].

Bibliographie

[1] J. Alon and S. Sclaroff. An integrated approach for segmentation and estimation of planar structure. Technical report 2000-23, Boston University, December 2000.

[2] M.-A. Ameller, A. Bartoli, and L. Quan. Minimal metric structure and motion from three affine images. In *Proceedings of the Fifth Asian Conference on Computer Vision, Melbourne, Australia*, volume I, pages 356–361, January 2002.

[3] M.-A. Ameller, A. Bartoli, and L. Quan. Reconstruction métrique minimale à partir de trois caméras affines. In *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, volume 2, pages 471–477, January 2002.

[4] N. Andreff, B. Espiau, and R. Horaud. Visual servoing from lines. In *International Conference on Robotics and Automation*, San Francisco, April 2000.

[5] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d points sets. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 9 :698–700, 1987.

[6] K.B. Atkinson, editor. *Close Range Photogrammetry and Machine Vision*. Whittles Publishing, 1996.

[7] S. Avidan and A. Shashua. Trajectory triangulation : 3D reconstruction of moving points from a monocular image sequence. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 22(4) :348–357, April 2000.

[8] N. Ayache. *Vision stéréoscopique et perception multisensorielle : application à la robotique mobile*. Science Informatique. InterEditions, 1989.

[9] C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA*, pages 559–565. IEEE Computer Society Press, June 1999.

[10] A. Bartoli. Piecewise planar segmentation for automatic scene modeling. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume II, pages 283–289. IEEE Computer Society Press, December 2001.

[11] A. Bartoli. Segmentation en plans pour la modélisation automatique à partir d'images. In *Actes de la journée thématique "Coopération Analyse d'Image et Modélisation", Lyon, France*, pages 37–40, June 2001.

[12] A. Bartoli. The geometry of dynamic scenes – on coplanar and convergent linear motions embedded in 3D static scenes. In *Proceedings of the 13th British Machine Vision Conference, Cardiff, England*, September 2002.

[13] A. Bartoli. On the non-linear optimization of projective motion using minimal parameters. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 2, pages 340–354, May 2002.

[14] A. Bartoli. A unified framework for quasi-linear bundle adjustment. In *Proceedings of the 16th* IAPR *International Conference on Pattern Recognition, Quebec City, Canada*, volume II, pages 560–563, August 2002.

[15] A. Bartoli. Towards gauge invariant bundle adjustment : A solution based on gauge dependent damping. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*. IEEE Computer Society Press, October 2003. to appear.

[16] A. Bartoli, N. Dalal, B. Bose, and R. Horaud. From video sequences to motion panoramas. In *Proceedings of the* IEEE *workshop on Motion and Video Computing, Orlando, Florida, USA*, 2002.

[17] A. Bartoli, N. Dalal, and R. Horaud. Des séquences vidéo aux panoramas de mouvement. In *CORESA'03 - actes des journées "COmpression et REprésentation des Signaux Audiovisuels", Lyon, France*, pages 171–174, January 2003.

[18] A. Bartoli, N. Dalal, and R. Horaud. Motion panoramas. Research Report 4771, INRIA, Grenoble, France, March 2003.

[19] A. Bartoli, R. Hartley, and F. Kahl. Motion from 3D line correspondences : Linear and non-linear solutions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume I, pages 477–484. IEEE Computer Society Press, June 2003.

[20] A. Bartoli and P. Sturm. The 3D line motion matrix and alignment of line reconstructions. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume I, pages 287–292. IEEE Computer Society Press, December 2001.

[21] A. Bartoli and P. Sturm. Constrained structure and motion from N views of a piecewise planar scene. In *Proceedings of the First International Symposium on Virtual and Augmented Architecture, VAA'01, Dublin, Ireland*, pages 195–206, June 2001.

[22] A. Bartoli and P. Sturm. Three new algorithms for projective bundle adjustment with minimum parameters. Research Report 4236, INRIA, Grenoble, France, August 2001.

[23] A. Bartoli and P. Sturm. La matrice de mouvement pour droites 3D, application à l'alignement de reconstructions de droites. In *13ème Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle*, volume I, pages 29–37, January 2002.

[24] A. Bartoli and P. Sturm. The 3D line motion matrix and alignment of line reconstructions. *International Journal of Computer Vision*, 2003. to appear.

[25] A. Bartoli and P. Sturm. Constrained structure and motion from multiple uncalibrated views of a piecewise planar scene. *International Journal of Computer Vision*, 52(1) :45–64, April 2003.

[26] A. Bartoli and P. Sturm. Multiple-view structure and motion from line correspondences. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*. IEEE Computer Society Press, October 2003. to appear.

[27] A. Bartoli and P. Sturm. Non-linear estimation of the fundamental matrix with minimal parameters. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2003. to appear.

[28] A. Bartoli, P. Sturm, and R. Horaud. A projective framework for structure and motion recovery from two views of a piecewise planar scene. Research Report 4070, INRIA, Grenoble, France, October 2000.

[29] A. Bartoli, P. Sturm, and R. Horaud. Projective structure and motion from two views of a piecewise planar scene. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, volume 1, pages 593–598, July 2001.

[30] A. Bartoli, P. Sturm, and R. Horaud. Structure and motion from two uncalibrated views using points on planes. In *Proceedings of the Third International Conference on 3D Digital Imaging and Modeling, 3DIM'01, Québec City, Canada*, pages 83–90, May 2001.

[31] A. Bartoli, P. Sturm, and R. Horaud. Triangulation projective contrainte par multi-coplanarité. In *Journées ORASIS 2001, Cahors, France*, pages 47–56, June 2001.

[32] P. Beardsley, P. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In B. Buxton and R. Cipolla, editors, *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1065 of *Lecture Notes in Computer Science*, pages 683–695. Springer-Verlag, April 1996.

[33] R. Berthilsson and A. Heyden. Recognition of planar point configurations using the density of affine shape. In *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, pages 72–88, 1998.

[34] R.H. Boggs, J.R. Donaldson, and R.B. Schnabel. ODRPACK : software for weighted orthogonal distance regression. *ACM Transaction on Mathematical Software*, 15(4) :348–364, 1989.

[35] D. Bondyfalat and S. Bougnoux. Imposing euclidean constraints during self-calibration processes. In R. Koch and L. Van Gool, editors, *Proceedings of the SMILE Workshop on 3D Structure from Multiple Images of Large-Scale Environments, Freiburg, Germany*, volume 1506 of *Lecture Notes in Computer Science*, pages 224–235. Springer-Verlag, June 1998.

[36] S. Bougnoux. From projective to euclidean space under any practical situation, a criticism of self-calibration. In *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, pages 790–796, 1998.

[37] J.-Y. Bouguet and P. Perona. Motion from points, lines and P-lines on N views. Technical report, California Institute of Technology, 1996.

[38] T. Buchanan. Critical sets for 3D reconstruction using lines. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 730–738. Springer-Verlag, May 1992.

[39] J. Canny. A computational approach to edge detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(6) :679–698, 1986.

[40] S. Carlsson. Duality of reconstruction and positioning from projective views. In *Workshop on Representation of Visual Scenes, Cambridge, Massachusetts, USA*, pages 85–92, June 1995.

[41] W. Chojnacki, M.J. Brooks, A. van den Hengel, and D. Gawley. Revisiting hartley's normalised eight-point algorithm. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 2003. to appear.

[42] O. Chum and T. Pajdla. The geometric error for homography. Technical Report 27, Czech Technical University, Prague, CMP, 2001.

[43] M.A.R. Cooper and S. Robson. Theory of close range photogrammetry. In K.B. Atkinson, editor, *Close Range Photogrammetry and Machine Vision*, chapter 2, pages 9–51. Whittles Publishing, Roseleigh House, Latheronwheel, Caithness, KW5 6DW, Scotland, UK., 1996.

[44] S. Cornou, M. Dhome, and P. Sayd. Reconstruction de bâtiments à partir de N vues non calibrées. In *Journées ORASIS 2003, Gérardmer, France*, pages 51–57, May 2003.

[45] J. Costeira and T. Kanade. A multi-body factorization method for motion analysis. In E. Grimson, editor, *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 1071–1076. IEEE, IEEE Computer Society Press, June 1995.

[46] H. Crapo and R. Penne. Cheirality and the isotropy classification of skew lines in projective 3-space. *Advances in Mathematics*, 103(1) :1–106, 1994.

[47] G. Csurka, D. Demirdjian, and R. Horaud. Finding the collineation between two projective reconstructions. *Computer Vision and Image Understanding*, 75(3) :260–268, September 1999.

[48] G. Csurka and O.D. Faugeras. Computing three-dimensional projective invariants from a pair of images using the Grassmann-Cayley algebra. In *Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision, Xian, China*, pages 150–157, 1995.

[49] G. Csurka, C. Zeller, Z. Zhang, and O. Faugeras. Characterizing the uncertainty of the fundamental matrix. *Computer Vision and Image Understanding*, 1997.

[50] P.E. Debevec, C.J. Taylor, and J. Malik. Modeling and rendering architecture from photographs : a hybrid geometry-and image-based approach. In SIGGRAPH '96, *New Orleans*, August 1996.

[51] D. Demirdjian and R. Horaud. Motion-egomotion discrimination and motion segmentation from image-pair streams. *Computer Vision and Image Understanding*, 78(1) :53–68, April 2000.

[52] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 20(3) :720–755, 1999.

[53] A. Dermanis. The photogrammetric inner constraints. ISPRS *Journal of Photogrammetry and Remote Sensing*, 49(1) :25–39, 1994.

[54] A. R. Dick, P.H.S. Torr, and R. Cipolla. Automatic 3D modelling of architecture. In *The Eleventh British Machine Vision Conference, University of Bristol, UK*, 2000.

[55] A.R. Dick, P.H.S. Torr, S.F. Ruffle, and R. Cipolla. Combining single view recognition and multiple view stereo for architectural scenes. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, 2001.

[56] N. Ayache et Faugeras. Maintaining representations of the environment of a mobile robot. IEEE *Transactions on Robotics and Automation*, 5(6) :804–819, 1989.

[57] O. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig ? In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 563–578. Springer-Verlag, May 1992.

[58] O. Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint*. Artificial intelligence. The MIT Press, Cambridge, MA, USA, Cambridge, MA, 1993.

[59] O. Faugeras and M. Hebert. The representation, recognition, and locating of 3D objects. *The International Journal of Robotics Research*, 5 :27–52, 1986.

[60] O. Faugeras, Q.-T. Luong, and T. Papadopoulo. *The Geometry of Multiple Images*. MIT Press, March 2001.

[61] O. Faugeras and F. Lustman. Motion and structure from motion in a piecewise planar environment. *International Journal of Pattern Recognition and Artificial Intelligence*, 2(3) :485–508, September 1988.

[62] O. Faugeras and B. Mourrain. On the geometry and algebra of the point and line correspondences between n images. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 951–956, June 1995.

[63] O.D. Faugeras and G. Toscani. Camera calibration for 3D computer vision. In *Proceedings of International Workshop on Machine Vision and Machine Intelligence, Tokyo, Japan*, 1987.

[64] M.A. Fischler and R.C. Bolles. Random sample consensus : A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6) :381 – 395, June 1981.

[65] A. Fitzgibbon. Simultaneous linear estimation of multiple view geometry and lens distortion. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, December 2001.

[66] A.W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *European Conference on Computer Vision*, pages 311–326, june 1998.

[67] A.W. Fitzgibbon and A. Zisserman. Multibody structure and motion : 3-D reconstruction of independently moving objects. In D. Vernon, editor, *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, volume 1842 of *Lecture Notes in Computer Science*, pages 891–906. Springer-Verlag, June 2000.

[68] P. Fornland and C. Schnörr. A robust and convergent iterative approach for determining the dominant plane from two views without correspondence and calibration. In IEEE, editor, *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 508–513, 1997.

[69] W. Förstner. Algebraic projectif geometru and direct optimal estimation of geometric entities. In AAPR *ÖAGM conference, Graz, Austria*, 2001.

[70] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*. Academic Press, 1981.

[71] G.H. Golub and C.F. van Loan. *Matrix Computation*. The Johns Hopkins University Press, Baltimore, 1989.

[72] V. M. Govindu. Combining two-view constraints for motion estimation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, December 2001.

[73] N. Guilbert and A. Bartoli. Batch recovery of multiple views with missing data using direct sparse solvers. In *Proceedings of the 14th British Machine Vision Conference, Norwich, England*, September 2003. to appear.

[74] P. Gurdjos, A. Crouzil, and R. Payrissat. Another way of looking at plane-based calibration : the centre circle constraint. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, May 2002.

[75] A. Habib. Motion parameter estimation by tracking stationary three-dimensional straight lines in image sequences. *International Archives of Photogrammetry and Remote Sensing*, 53, 1998.

[76] A. Habib, M. Morgan, and Y.-R. Lee. Bundle ajustement with self-calibration using straight lines. *Photogrammetric Record*, October 2002.

[77] G. Hager and K. Toyama. X vision : A portable substrate for real-time vision applications. *CVIU*, 69(1) :23–37, 1998.

[78] M. Han and T. Kanade. Reconstruction of a scene with multiple linearly moving objects. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, June 2000.

[79] M. Han and T. Kanade. Multiple motion scene reconstruction from uncalibrated views. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, July 2001.

[80] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–151, 1988.

[81] R. Hartley. Invariants of lines in space. In *Proceedings of* DARPA *Image Understanding Workshop*, pages 737–744, 1993.

[82] R. Hartley. In defence of the 8-point algorithm. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 1064–1070, June 1995.

[83] R. Hartley. Minimizing algebraic error. In *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, 1998.

[84] R. Hartley and F. Kahl. A critical configuration for reconstruction from rectilinear motion. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume I, pages 511–517. IEEE Computer Society Press, June 2003.

[85] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2) :146–157, 1997.

[86] R. I. Hartley. In defence of the eight-point algorithm. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(6) :580–593, June 1997.

[87] R. I. Hartley and G. Debunne. Dualizing scene reconstruction algorithms. In *Proceedings of the SMILE Workshop on 3D Structure from Multiple Images of Large-Scale Environments, Freiburg, Germany*, pages 14–31, 1998.

[88] R.I. Hartley. Estimation of relative camera positions for uncalibrated cameras. In G. Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision, Santa Margherita Ligure, Italy*, pages 579–587. Springer-Verlag, 1992.

[89] R.I. Hartley. Euclidean reconstruction from uncalibrated views. In *Proceeding of the* DARPA–ESPRIT *workshop on Applications of Invariants in Computer Vision, Azores, Portugal*, pages 187–202, October 1993.

[90] R.I. Hartley. Projective reconstruction and invariants from multiple images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 16(10) :1036–1041, October 1994.

[91] R.I. Hartley. Kruppa's equations derived from the fundamental matrix. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(2) :133–135, February 1997.

[92] R.I. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2) :125–140, 1997.

[93] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000.

[94] J. Heikkinen. Linear feature based approach to map revision. *International Archives of Photogrammetry and Remote Sensing*, 30, part 4 :344–351, 1994.

[95] O. Henricsson, A. Streilein, and A. Gruen. Automated 3-D reconstruction of buildings and visualization of city models. In *Proceedings of the Workshop on 3D City Models, Bonn, Germany*, October 1996.

[96] A. Heyden. Reduced multilinear constraints - theory and experiments. *International Journal of Computer Vision*, 1(30) :5–26, 1998.

[97] A. Heyden and K. Åström. A canonical framework for sequences of images. In *Workshop on Representation of Visual Scenes, Cambridge, Massachusetts, USA*, pages 45–52, June 1995.

[98] A. Heyden and K. Åström. Algebraic properties of multilinear constraints. *Mathematical Methods in the Applied Sciences*, 20 :1135–1162, 1997.

[99] A. Heyden and F. Kahl. Reconstruction from affine cameras using closure constraints. In *Proceedings of the 14th International Conference on Pattern Recognition, Brisbane, Australia*, pages 47–50, 1998.

[100] R. Horaud, G. Csurka, and D. Demirdjian. Stereo calibration from rigid motions. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 22(12) :1446–1452, December 2000.

[101] B.K.P. Horn. Closed form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4) :629–642, 1987.

[102] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7) :1127–1135, July 1988.

[103] M. Irani and P. Anandan. Factorization with uncertainty. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, pages 539–553, 2000.

[104] D. Jacobs. Linear fitting with missing data : Applications to structure-from-motion and to characterizing intensity images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 206–212. IEEE Computer Society Press, June 1997.

[105] F. Jurie and M. Dhome. Hyperplane approximation for template matching. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24(7) :996–1000, 2002.

[106] F. Kahl and R. Hartley. Critical curves and surfaces for euclidean reconstruction. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, June 2002.

[107] F. Kahl, R. Hartley, and K. Åström. Critical configurations for N-view projective reconstruction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, December 2001.

[108] F. Kahl, B. Triggs, and K. Åström. Critical motions for auto-calibration when some intrinsic parameters can vary. *Journal of Mathematical Imaging and Vision*, 13(2) :131–146, October 2000.

[109] B. Kamgar-Parsi and B. Kamgar-Parsi. An open problem in matching sets of 3D lines. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume I, pages 651–656, December 2001.

[110] K. Kanatani. *Statistical Optimisation for Geometric Computation : Theory and Pra*. Elsevier Science, 1996.

[111] K. Kanatani. Geometric information criterion for model selection. *International Journal of Computer Vision*, 26(3) :171–189, 1998.

[112] K. Kanatani and D. D. Morris. Gauges and gauge transformations for uncertainty description of geometric structure with indeterminacy. IEEE *Transactions on Information Theory*, 47(5), July 2001.

[113] R. Koch. Surface segmentation and modeling of 3D polygonal objects from stereoscopic image pairs. In *Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria*, 1996.

[114] K. Kubik. Relative and absolute orientation based on linear features. ISPRS *Journal of Photogrammetry and Remote Sensing*, 46(1) :407–417, 1991.

[115] K.N. Kutulakos. Approximate N-view stereo. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, 2000.

[116] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 2000.

[117] F. Lang and W. Förstner. 3D-city modeling with a digital one-eye stereo system. In *Proceedings of the XVIII* ISPRS-*Congress, Vienna, Austria*, July 1996.

[118] J.-M. Lavest, M. Viala, and M. Dhome. Do we really need an accurate calibration pattern to achieve a reliable camera calibration ? In *Proceedings of the 5th European Conference on Computer Vision, Freiburg, Germany*, volume 1, pages 158–174, 1998.

[119] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. SIAM Classics In Applied Mathematics, 1995.

[120] Y. Leedan and P. Meer. Heteroscedastic regression in computer vision : Problems with bilinear constraints. *International Journal of Computer Vision*, 37(2) :127–150, 2000.

[121] K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, pages 164–168, 1944.

[122] D. Liebowitz and A. Zisserman. Metric rectification for perspective images of planes. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Santa Barbara, California, USA*, pages 482–488, June 1998.

[123] Y. Liu and T.S. Huang. Estimation of rigid body motion using straight line correspondences. *Computer Vision, Graphics and Image Processing*, 43(1) :37–52, July 1988.

[124] Y. Liu and T.S. Huang. A linear algorithm for motion estimation using straight line correspondences. *Computer Vision, Graphics and Image Processing*, 44(1) :35–57, October 1988.

[125] Y. Liu, T.S. Huang, and O.D. Faugeras. Determination of camera location from 2D to 3D line and point correspondences. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 12(1) :28–37, January 1990.

[126] H.C. Longuet-Higgins. A computer program for reconstructing a scene from two projections. *Nature*, 293 :133–135, September 1981.

[127] H.C. Longuet-Higgins. A method of obtaining the relative positions of 4 points from 3 perspective projections. In *Proceedings of the British Machine Vision Conference, Glasgow, Scotland*, pages 86–94, 1991.

[128] Q.T. Luong and O. Faugeras. The fundamental matrix : Theory, algorithms and stability analysis. *International Journal of Computer Vision*, 17(1) :43–76, 1996.

[129] Q.T. Luong and O.D. Faugeras. Determining the fundamental matrix with planes : Instability and new algorithms. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, New York, USA*, pages 489–494, June 1993.

[130] Q.T. Luong and T. Vieville. Canonic representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2) :193–229, 1996.

[131] E. Malis and A. Bartoli. Euclidean bundle adjustment independent on camera intrinsic parameters. Research Report 4377, INRIA, Grenoble, France, December 2001.

[132] E. Malis and R. Cipolla. Camera self-calibration from unknown planar structures enforcing the multi-view constraints between collineations. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 4(9), September 2002.

[133] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2) :431–441, June 1963.

[134] D. Martinec and T. Padjla. Structure from many perspective images with occlusions. In *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, pages 355–369, 2002.

[135] D. Martinec and T. Pajdla. Line reconstruction from many perspective images by factorization. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume I, pages 497–502. IEEE Computer Society Press, June 2003.

[136] B. Matei and P. Meer. Optimal rigid motion estimation and performance evaluation with bootstrap. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Fort Collins, Colorado, USA*, pages 339–345, 1999.

[137] S.J. Maybank and O.D. Faugeras. A theory of self calibration of a moving camera. *International Journal of Computer Vision*, 8(2) :123–151, 1992.

[138] C. McGlone. Bundle adjustment with object space geometric constraints for site modeling. In D.M. McKeown and Dowman, editors, *Proceedings of the* SPIE *Conference on Integrating Photogrammetric Techniques with Scene Analysis and Machine Vision II, Orlando, Florida, USA*, volume 2486, pages 25–36, April 1995.

[139] P. McLauchlan, X. Shen, A. Manessis, P. Palmer, and A. Hilton. Surface-based structure-from-motion using feature groupings. In *Proceedings of the Fourth Asian Conference on Computer Vision*, 2000.

[140] P. F. McLauchlan. Gauge invariance in projective 3D reconstruction. In *Proceedings of the Multi-View Workshop, Fort Collins, Colorado, USA*, 1999.

[141] P. F. McLauchlan. Gauge independence in optimization algorithms for 3D vision. In *Proceedings of the Vision Algorithms Workshop, Dublin, Ireland*, 2000.

[142] P. F. McLauchlan and D. W. Murray. A unifying framework for structure and motion recovery from image sequences. In *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, June 1995.

[143] P. Meer, D. Mintz, A. Rosenfeld, and D.Y. Kim. Robust regression methods for computer vision : a review. *International Journal of Computer Vision*, 6(1) :59–70, 1991.

[144] E. M. Mikhail. Relative control for extraterrestrial work. *Photogrammetric Engineering*, 4(36), m4 1970.

[145] R. Mohr, L. Quan, and F. Veillon. Relative 3D reconstruction using multiple uncalibrated images. *The International Journal of Robotics Research*, 14(6) :619–632, 1995.

[146] J. J. Moré. The levenberg-marquardt algorithm : Implementation and theory. In G. Watson, editor, *Numerical Analysis*, number 630 in Lecture Notes in Mathematics, pages 105–116. Springer-Verlag, 1978.

[147] D.D. Morris and T. Kanade. A unified factorization algorithm for points, line segments and planes with uncertainty models. In *Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, pages 696–702, January 1998.

[148] D.D. Morris and T. Kanade. Image-consistent surface triangulation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, 2000.

[149] M. Mühlich and R. Mester. The role of total least squares in motion analysis. In *Proceedings of the 5th European Conference on Computer Vision, Freiburg, Germany*, pages 305–321, June 1998.

[150] D. C. Mulawa and E. M. Mikhail. Photogrammetric treatment of linear features. *International Archives of Photogrammetry and Remote Sensing*, 27 :383–393, 1988.

[151] N. Navab, B. Bascle, M. Loser, B. Geiger, and R. Taylor. Visual servoing for automatic and uncalibrated needle placement for percutaneaous procedures. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, pages 327–334, 2000.

[152] N. Navab and O. D. Faugeras. The critical sets of lines for camera displacement estimation : a mixed euclidean-projective and constructive approach. *International Journal of Computer Vision*, 23(1) :17–44, 1997.

[153] N. Navab, Y. Genc, and M. Appel. Lines in one orthographic and two perspective views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, volume II, pages 607–614, 2000.

[154] N. Navab and Z. Zhang. Fusion of visual data through dynamic stereo-motion cooperation. In *Proceedings of the XVII* ISPRS*-Congress, Washington, USA*, pages 932–939, 1992.

[155] D. Nistèr. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, volume 1, pages 649–663, 2000.

[156] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer-Verlag, 1999.

[157] T. Okatani and K. Deguchi. Toward a statistically optimal method for estimating geometric relation from noisy data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume I, pages 432–439. IEEE Computer Society Press, June 2003.

[158] T. Papadopoulo and M.I.A. Lourakis. Estimating the Jacobian of the singular value decomposition : Theory and applications. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, volume I, pages 554–570, 2000.

[159] J. Plücker. On a new geometry of space. *Philosophical Transactions of the Royal Society of London A*, 155 :725–791, 1865.

[160] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. *International Journal of Computer Vision*, 32(1) :7–25, 1999.

[161] H. Pottmann and J. Wallner. *Computational line geometry*. Springer-Verlag, Berlin, Germany, 2001.

[162] B. Prescott and G. I. McLean. Line based correction of radial lens distorsion. *Computer Vision, Graphics and Image Processing : Graphical Models and Image Processing*, 59(1) :39–47, 1997.

[163] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[164] L. Quan and T. Kanade. Affine structure from line correspondences with uncalibrated affine cameras. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 19(8) :834–845, August 1997.

[165] I.D. Reid and D.W. Murray. Active tracking of foveated feature clusters using affine structure. *International Journal of Computer Vision*, 18(1) :41–60, April 1996.

[166] K. Roberts. A new representation for a line. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, California, USA*, pages 635–640, 1988.

[167] T. Rodriguez, P. Sturm, M. Wilczkowiak, A. Bartoli, M. Personnaz, N. Guilbert, F. Kahl, M. Johansson, A. Heyden, J. M. Menendez, J. I. Ronda, and F. Jaureguizar. VISIRE. photorealistic 3D reconstruction from video sequences. In *Proceedings of the* IEEE *International Conference on Image Processing, Barcelona, Spain*, September 2003.

[168] R. A. Romano. *Projective minimal analysis of camera geometry*. Thèse de doctorat d'etat, MIT, May 2002.

[169] C. Rother. *Multi-view reconstruction and camera recovery using a real or virtual reference plane*. PhD thesis, KTH, Stockholm, Sweden, 2003.

[170] F. Schaffalitzky and A. Zisserman. Planar grouping for automatic detection of vanishing lines and points. *Image and Vision Computing*, 18(9) :647–658, 2000.

[171] F. Schaffalitzky, A. Zisserman, R. Hartley, and P.H.S. Torr. A six point solution for structure and motion. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*. Springer-Verlag, 2000.

[172] C. Schmid, R. Mohr, and C. Bauckhage. Evaluation of interest point detectors. *International Journal of Computer Vision*, 37(2) :151–172, 2000.

[173] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3) :199–234, 2000.

[174] G. A. F. Seber and C. J. Wild. *Non-linear regression*. John Wiley & Sons, New York, 1989.

[175] Y. Seo and K. S. Hong. Sequential reconstruction of lines in projective space. In *Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria*, pages 503–507, August 1996.

[176] Y. Shan, Z. Liu, and Z. Zhang. Model-based bundle adjustment with application to face modeling. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, July 2001.

[177] A. Shashua. Trilinearity in visual recognition by alignment. In J.O. Eklundh, editor, *Proceedings of the 3rd European Conference on Computer Vision, Stockholm, Sweden*, pages 479–484. Springer-Verlag, May 1994.

[178] A. Shashua. Trilinear tensor : The fundamental construct of multiple-view geometry and its applications. In *International Workshop on Algebraic Frames For The Perception Action Cycle (AFPAC), Kiel Germany*, pages 8–9, September 1997.

[179] A. Shashua and N. Navab. Relative affine structure : Canonical model for 3D from 2D geometry and applications. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 18(9) :873–883, 1996.

[180] A. Shashua and L. Wolf. Homography tensors : On algebraic entities that represent three views of static or moving planar points. In D. Vernon, editor, *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, volume 1842 of *Lecture Notes in Computer Science*, pages 507–521. Springer-Verlag, June 2000.

[181] A. Shashua and L. Wolf. On the structure and properties of the quadrifocal tensor. In *Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland*, pages 710–724, June 2000.

[182] D. Sinclair and A. Blake. Quantitative planar region detection. *International Journal of Computer Vision*, 18(1) :77–91, 1996.

[183] D. Sinclair, A. Blake, S. Smith, and C. Rothwell. Planar region detection and motion recovery. *Image and Vision Computing*, 11(4) :229–234, may 1993.

[184] D. Sinclair, H. Christensen, and C. Rothwell. Using the relation between a plane projectivity and the fundamental matrix. In *Proceedings of the 9th Scandinavian Conference on Image Analysis, Uppsala, Sweden*, 1995.

[185] C.C. Slama, editor. *Manual of Photogrammetry, Fourth Edition*. American Society of Photogrammetry and Remote Sensing, Falls Church, Virginia, USA, 1980.

[186] G. Sparr. Simultaneous reconstruction of scene structure and camera locations from uncalibrated image sequences. In *Proceedings of the 13th International Conference on Pattern Recognition, Vienna, Austria*, volume I, pages 328–333. IEEE Computer Society Press, August 1996.

[187] M. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4 :171–183, 1990.

[188] M. E. Spetsakis. A linear algorithm for point and line-based structure from motion. *Computer Vision, Graphics and Image Processing : Image Understanding*, 56(2) :230–241, September 1992.

[189] G. P. Stein and A. Shashua. On degeneracy of linear reconstruction from three views : Linear line complex and applications. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 21(3) :244–251, 1999.

[190] A. Streilein and U. Hirschberg. Integration of digital photogrammetry and CAAD : Constraint-based modeling and semi-automatic measurement. In *Proceedings of the International* CAAD *Futures Conference, Singapore*, September 1995.

[191] G. Strunz. Feature based image orientation and object reconstruction. *International Archives of Photogrammetry and Remote Sensing*, XXIX :113–118, 1992.

[192] J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6(4) :422–430, October 1964.

[193] P. Sturm. Critical motion sequences and conjugacy of ambiguous euclidean reconstructions. In M. Frydrych, J. Parkkinen, and A. Visa, editors, *Proceedings of the 10th Scandinavian Conference on Image Analysis, Lappeenranta, Finland*, volume I, pages 439–446, June 1997.

[194] P. Sturm. Critical motion sequences for monocular self-calibration and uncalibrated euclidean reconstruction. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 1100–1105, June 1997.

[195] P. Sturm. *Vision 3D non calibrée : contributions à la reconstruction projective et étude des mouvements critiques pour l'auto-calibrage*. Thèse de doctorat, Institut National Polytechnique de Grenoble, December 1997.

[196] P. Sturm. Critical motion sequences for the self-calibration of cameras and stereo systems with variable focal length. *Image and Vision Computing*, 20(5-6) :415–426, 2002.

[197] P. Sturm. Structure and motion for dynamic scenes – the case of points moving in planes. In A. Heyden, G. Sparr, M. Nielsen, and P. Johansen, editors, *Proceedings of the 7th European Conference on Computer Vision, Copenhagen, Denmark*, volume 2351 of *Lecture Notes in Computer Science*, pages 867–882, May 2002.

[198] P. Sturm and B. Triggs. A factorization based algorithm for multi-image projective structure and motion. In B. Buxton and R. Cipolla, editors, *Proceedings of the 4th European Conference on Computer Vision, Cambridge, England*, volume 1065 of *Lecture Notes in Computer Science*, pages 709–720. Springer-Verlag, April 1996.

[199] Z. Sun, A. Murat Tekalp, N. Navab, and V. Ramesh. Interactive optimization of 3D shape and 2D correspondences using multiple geometric constraints via POCS. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24(4) :562–569, 2002.

[200] R. Szeliski and P.H.S. Torr. Geometrically constrained structure from motion : Points on planes. In *3D Structure from Multiple Images of Large-scale Environments SMILE'98*. Springer Verlag, June 1998.

[201] J.-P. Tarel and J.-M. Vézien. A generic approach for planar patches stereo reconstruction. In *Proceedings of the Scandinavian Conference on Image Analysis*, pages 1061–1070, Uppsala, Sweden, 1995.

[202] C. J. Taylor and D. J. Kriegman. Minimization on the Lie group SO(3) and related manifolds. Research Report 9405, Yale University department of computer science, April 1994.

[203] C.J. Taylor and D.J. Kriegman. Structure and motion from line segments in multiple images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 17(11) :1021–1032, November 1995.

[204] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography : A factorization method. *International Journal of Computer Vision*, 9(2) :137–154, November 1992.

[205] A. Tommaselli and J. Lugnani. An alternative mathematical model to collinearity equations using straight features. *International Archives of Photogrammetry and Remote Sensing*, 27 :765–774, 1998.

[206] P. Torr and A. Zisserman. MLESAC : A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1), 2000.

[207] P.H.S. Torr. An assessment of information criteria for motion model selection. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 47–52, June 1997.

[208] P.H.S. Torr and D.W. Murray. Outlier detection and motion segmentation. In P.S. Schenker, editor, *Sensor Fusion VI*, pages 432–442, Boston, 1993. SPIE volume 2059.

[209] P.H.S. Torr and D.W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3) :271–300, 1997.

[210] P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. In R.B. Fisher and E. Trucco, editors, *Proceedings of the seventh British Machine Vision Conference, Edinburgh, Scotland*, volume 2, pages 655–664. British Machine Vision Association, September 1996.

[211] P.H.S. Torr and A. Zisserman. Robust parameterization and computation of the trifocal tensor. *Image and Vision Computing*, 15 :591–605, 1997.

[212] P.H.S. Torr, A. Zisserman, and S.J. Maybank. Robust detection of degenerate configurations while estimating the fundamental matrix. *Computer Vision and Image Understanding*, 71(3) :312–333, September 1998.

[213] B. Triggs. Matching constraints and the joint image. In E. Grimson, editor, *Proceedings of the 5th International Conference on Computer Vision, Cambridge, Massachusetts, USA*, pages 338–343. IEEE, IEEE Computer Society Press, June 1995.

[214] B. Triggs. Factorization methods for projective structure and motion. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, San Francisco, California, USA*, pages 845–851, 1996.

[215] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Puerto Rico, USA*, pages 609–614. IEEE Computer Society Press, June 1997.

[216] B. Triggs. Linear projective reconstruction from matching tensors. *Image and Vision Computing*, 15(8) :617–625, August 1997.

[217] B. Triggs. Autocalibration from planar scenes. In *Proceedings of the 5th European Conference on Computer Vision, Freiburg, Germany*, 1998.

[218] B. Triggs. Optimal estimation of matching constraints. In *3D Structure from Multiple Images of Large-scale Environments SMILE'98*. Springer Verlag, June 1998.

[219] B. Triggs, P.F. McLauchlan, R.I. Hartley, and A. Fitzgibbon. Bundle ajustment — a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Proceedings of the International Workshop on Vision Algorithms : Theory and Practice, Corfu, Greece*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

[220] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. IEEE *Journal of Robotics and Automation*, 3(4) :323–344, August 1987.

[221] F.A. van den Heuvel and G. Vosselman. Efficient 3D-modeling of buildings using a priori geometric object information. In *Proceedings of the* SPIE *Conference on Videometrics V*, volume 3174, pages 38–49, 1997.

[222] C. Vestri and F. Devernay. Using robust methods for automatic extraction of buildings. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, volume I, pages 133–138, December 2001.

[223] R. Vidal, Y. Ma, and S. Sastry. Generalized principal component analysis (GPCA). In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume I, pages 612–628. IEEE Computer Society Press, June 2003.

[224] R. Vidal and S. Sastry. Optimal segmentation of dynamic scenes from two perspective views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Madison, Wisconsin, USA*, volume II, pages 281–286. IEEE Computer Society Press, June 2003.

[225] T. Viéville, Q.T. Luong, and O.D. Faugeras. Motion of points and lines in the uncalibrated case. *International Journal of Computer Vision*, 17(1), 1995.

[226] J. Weng, T.S. Huang, and N. Ahuja. Motion and structure from line correspondences : Closed-form solution, uniqueness, and optimization. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 14(3) :318–336, 1992.

[227] M. Wilczkowiak, G. Trombettoni, C. Jermann, P. Sturm, and E. Boyer. Scene reconstruction based on constraint decomposition techniques. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, October 2003. to appear.

[228] L. Wolf and A. Shashua. On projection matrices $P^k \rightarrow P^2, k = 3, \cdots, 6$, and their applications in computer vision. In *Proceedings of the 8th International Conference on Computer Vision, Vancouver, Canada*, volume 1, pages 412–419. IEEE Computer Society Press, July 2001.

[229] L. Wolf and A. Shashua. Two-body segmentation from two perspective views. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Kauai, Hawaii, USA*, December 2001.

[230] L. Wolf, A. Shashua, and Y. Wexler. Join tensors : On 3D-to-3D alignment of dynamic sets. In A. Sanfeliu, J.J. Villanueva, M. Vanrell, R. Alquézar, J.-O. Eklundh, and Y. Aloimonos, editors, *Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain*, volume 1, pages 388–391, September 2000.

[231] G. Xu, J.-I. Terai, and H.-Y. Shum. A linear algorithm for camera self-calibration, motion and structure recovery for multi-planar scenes from two perspective images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition, Hilton Head Island, South Carolina, USA*, June 2000.

[232] L. Zelnik-Manor and M. Irani. Multi-view constraints on homographies. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 24(2) :214–223, February 2002.

[233] Z. Zhang. Determining the epipolar geometry and its uncertainty : A review. *International Journal of Computer Vision*, 27(2) :161–195, March 1998.

[234] Z. Zhang and O. D. Faugeras. Tracking and grouping 3D line segments. In *Proceedings of the 3rd International Conference on Computer Vision, Osaka, Japan*, pages p. 577–580, December 1990.

[235] Z. Zhang and O. D. Faugeras. Determining motion from 3D line segments : A comparative study. *Image and Vision Computing*, 9(1) :10–19, February 1991.

[236] Z. Zhang and C. Loop. Estimating the fundamental matrix by transforming image points in projective space. *Computer Vision and Image Understanding*, 82(2) :174–180, May 2001.

[237] A. Zisserman, P.A. Beardsley, and I.D. Reid. Metric calibration of a stereo rig. In *Workshop on Representation of Visual Scenes, Cambridge, Massachusetts, USA*, pages 93–100, June 1995.

[238] M. Zucchelli. *Optical Flow Based Structure from Motion*. PhD thesis, KTH, Stockholm, Sweden, 2002. ISBN 91-7283-308-4.