

Calibrage et Reconstruction à l'aide de Parallélépipèdes et de Parallélogrammes

Calibration and Reconstruction Using Parallelepipeds and Parallelograms

Marta Wilczkowiak* Peter Sturm Edmond Boyer

Movi-Gravir-Inria Rhône-Alpes, 655 Avenue de l'Europe, 38330 Montbonnot, France

Prenom.Nom@inrialpes.fr

Résumé. Dans cet article, nous nous intéressons à l'usage des parallélépipèdes et parallélogrammes dans les processus de calibrage et de reconstruction à partir d'images. Ces entités caractérisent de manière naturelle des contraintes géométriques fréquemment présentes dans les scènes d'environnements humains : le parallélisme et l'orthogonalité par exemple. Nous montrons en particulier que les caractéristiques intrinsèques d'une caméra sont complètement déterminées par celles d'un parallélépipède lorsqu'une image de ce parallélépipède est disponible, et inversement. Nous montrons que cette dualité entre les caractéristiques intrinsèques peut facilement être exploitée pour calibrer une caméra à partir de la projection d'un parallélépipède. Nous précisons de quelle manière utiliser cette relation lorsqu'une ou quelques images sont disponibles, comment en particulier combiner les connaissances sur les caméras et sur les primitives utilisées. Ce contexte théorique s'applique aisément à la reconstruction à partir d'images et à la réalité augmentée. Nous illustrons ces principes par des résultats pratiques sur des scènes pour lesquelles une ou quelques images non calibrées sont disponibles.

Abstract. In this paper, efficient tools for reconstruction from a single or a few images are presented. They exploit geometric constraints frequently present in man-made environments and allow camera calibration as well as scene structure to be estimated with few interactions and little a priori knowledge. The proposed approach is based on primitives that naturally characterize rigidity constraints: parallelepipeds and parallelograms. The intrinsic metric characteristics of these primitives appear to be dual to the intrinsic characteristics of a perspective camera. Thus any knowledge on either a primitive or a camera can be used to estimate the respective dual entity. We describe the constraints that can be derived in the single image context as well as when several images are available. These principles are illustrated by examples of reconstructions and augmented reality using uncalibrated images.

1 Introduction

Les environnements humains sont souvent constitués d'entités géométriques possédant des propriétés de parallélisme : bâtiments, fenêtres, etc. Ces entités représentent une source d'information importante sur la scène observée et sur la caméra utilisée lorsqu'une image de la scène est considérée. Leur utilisation dans les processus de modélisation à partir d'images permet de s'affranchir du besoin d'un grand nombre d'images, une seule pouvant souvent suffire. En outre, ces entités permettent de contraindre les reconstructions effectuées et donc d'obtenir des modèles d'un niveau de réalisme élevé. Dans cet article, nous étudions comment utiliser ces primitives pour le calibrage de caméra et la reconstruction à partir d'images. La motivation est de fournir des outils de modélisation simples, qui ne nécessitent que peu d'interventions humaines et permettent d'obtenir rapidement un modèle.

Les contraintes géométriques découlant de structures parallélépipédiques ont été largement utilisées en vision par ordinateur par le passé. La plupart des approches font usage des points de fuite que définissent les projections de côtés parallèles. Caprile et Torre [2] ont utilisé des parallélépipèdes rectangles. Les contraintes d'orthogonalité associées aux points de fuite fournissent en effet des informations sur les caractéristiques intrinsèques de la caméra utilisée. Suivant ce principe, plusieurs approches ont par la suite été proposées [14, 4, 16, 5, 17, 13]. Cependant, la détermination des points de fuite s'avère être un processus numériquement instable. De plus, ce type d'approche ne permet pas d'exploiter pleinement les informations fournies par une primitive telle qu'un parallélépipède, primitive qui est souvent à l'origine des points de fuites utilisés. L'approche proposée dans cet article repose au contraire sur l'identification précise des contraintes que fournissent les projections de parallélépipèdes et de parallélogrammes et sur l'étude exhaustive de leurs mises en œuvre pour le calibrage et la reconstruction.

Des approches basées sur le calcul d'homographies à partir de primitives coplanaires ont également été proposées

*M. Wilczkowiak est financée par le projet IST-1999-10756, VISIRE.

récemment [21, 23]. Bien que plus flexibles que les approches de calibrage classiques, elles nécessitent des informations Euclidiennes sur les primitives utilisées qui ne sont pas nécessairement disponibles. L'approche que nous proposons s'applique avec un éventail d'informations *a priori* plus large.

Citons aussi l'approche populaire de Debevec *et al.* [7] qui utilise plusieurs types de primitives dont les parallélépipèdes rectangles, mais qui nécessite plusieurs images ainsi que plusieurs connaissances *a priori* sur la caméra. Enfin, plus proche des travaux présentés dans cet article, Chen *et al.* [3] utilisent aussi les parallélépipèdes pour le calibrage dans le contexte de la réalité augmentée. Néanmoins, l'approche proposée reste limitée puisque la caméra doit satisfaire une contrainte forte (facteur d'échelle unitaire) et que seules des connaissances partielles sur les parallélépipèdes (les angles) peuvent être utilisées.

L'approche proposée dans cet article permet de calibrer, de reconstruire et d'incruster des modèles tridimensionnels à partir d'une ou de quelques images quelconques. La principale contribution est de montrer comment combiner des informations sur les caméras et sur de simples primitives (parallélépipèdes et parallélogrammes) pour calibrer et reconstruire. La méthode proposée repose sur la dualité existant entre les caractéristiques intrinsèques d'une caméra et celles des primitives citées dans le contexte des projections perspectives.

2 Préliminaires

Le modèle de caméra. Nous utilisons le modèle de caméra à *sténopé* (projection perspective). Dans ce modèle, la projection p dans le plan image du point P de l'espace tridimensionnel est décrite par la relation projective : $p \sim M \cdot P$, où \sim représente la relation d'égalité à un facteur d'échelle près et M est une matrice de projection 3×4 . La matrice M peut être décomposée en : $M = K \cdot [R t]$, où $[R t]$ est la matrice 3×4 déterminant l'orientation relative R^T et la position $-R^T t$ de la caméra dans l'espace et K est la matrice des paramètres intrinsèques de la caméra :

$$K = \begin{bmatrix} \alpha_u & s = 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1)$$

où α_u et α_v représentent la distance focale de la caméra exprimée suivant les tailles horizontale et verticale du pixel, s le facteur d'obliquité, considéré ici comme nul (mais cette contrainte peut être relâchée) et (u_0, v_0) les coordonnées en pixels de l'intersection de l'axe optique avec le plan image (point principal).

Les parallélépipèdes. Un parallélépipède est défini par douze paramètres : six paramètres externes décrivant son orientation et sa position dans l'espace, et six paramètres internes décrivant ses caractéristiques Euclidiennes – trois longueurs d'arêtes et trois angles entre les directions des arêtes (voir la figure 1). La projection perspective d'un parallélépipède est définie par onze paramètres. En consé-

quence, cinq points et une direction du plan image sont suffisants pour caractériser l'image d'un parallélépipède. Les projections des autres sommets du parallélépipède peuvent alors être déterminées par une simple règle perspective à trois points de fuite, en utilisant le parallélisme des arêtes du parallélépipède (voir la figure 2). Il en résulte que le parallélisme ne fournit aucune contrainte sur la projection et n'est donc pas utile pour le calibrage. Ce dernier est basé sur la connaissance d'informations Euclidiennes sur le parallélépipède : les rapports de longueurs et les angles entre arêtes. Les projections additionnelles de sommets (en plus des cinq et demi citées) sont utiles pour stabiliser numériquement les résultats, mais ne fournissent pas de contraintes nouvelles pour le calibrage.

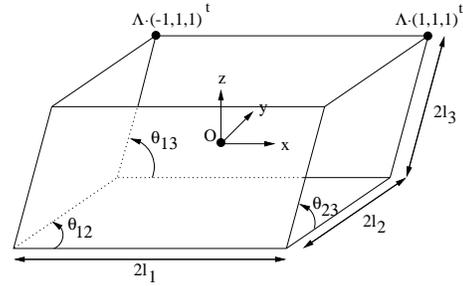


FIG. 1 – Le paramétrage du parallélépipède : $2l_i$ (les longueurs des arêtes) ; θ_{ij} (les angles entre les arêtes du parallélépipède).

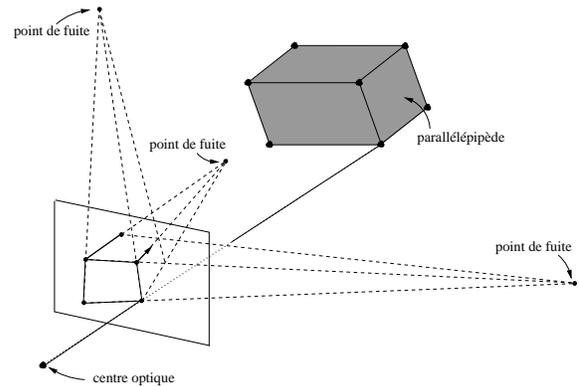


FIG. 2 – Un parallélépipède et sa projection qui est définie uniquement par cinq points et une direction dans l'image. Les projections des arêtes du parallélépipède définissent trois points de fuite dans l'image.

Les parallélogrammes. Un parallélogramme peut être considéré comme l'équivalent 2D d'un parallélépipède et peut être décrit de façon similaire : six paramètres externes pour son orientation et sa position dans l'espace 3D et trois paramètres internes pour ses caractéristiques Euclidiennes – deux longueurs d'arêtes et un angle entre les arêtes. La transformation (une homographie) entre le plan contenant un parallélogramme et le plan image dépend de huit paramètres. Les projections de quatre points constituant les sommets d'un parallélogramme sont donc suffisantes pour estimer cette transformation.

3 Dualité des paramètres intrinsèques

Les méthodes classiques utilisant les contraintes de parallélisme et d'orthogonalité consistent habituellement à déterminer les points de fuite, puis à exploiter le fait que les directions correspondantes dans l'espace sont orthogonales [2, 13]. Les points de fuite sont généralement estimés en cherchant des intersections de droites dans l'image, ou, plus précisément, les points qui minimisent la distance aux projections des droites parallèles [2, 15, 16]. De telles méthodes sont sujettes à de fortes instabilités numériques, en particulier lorsque les arêtes projetées dans l'image sont quasi-parallèles.

Contrairement à ces méthodes, nous ne calculons pas directement les points de fuite, mais nous estimons la meilleure matrice de projection telle que les points définis dans l'image correspondent aux projections des sommets d'un parallélépipède. Nous appelons cette matrice *matrice de projection canonique*. À partir de cette matrice, nous pouvons établir les relations existant entre les paramètres intrinsèques du parallélépipède, ou du parallélogramme, et celles de la caméra. Chaque information sur une entité donne alors une contrainte sur l'entité duale. Cela peut être utilisé pour le calcul des paramètres de la caméra (le calibrage) et du parallélépipède/parallélogramme (reconstruction). Dans la suite, nous établissons les relations de dualité puis nous montrons (partie 4) comment utiliser cette relation avec une ou plusieurs images.

3.1 Matrices de projection canoniques

La matrice de projection canonique d'un parallélépipède. Soit un parallélépipède dont les paramètres sont R , $-R^T t$, (l_1, l_2, l_3) et $(\theta_{12}, \theta_{23}, \theta_{13})$ – les paramètres définissant, respectivement, son orientation, sa position¹, ses longueurs et angles. On peut associer à ce parallélépipède un repère affine canonique, dans lequel le parallélépipède est défini comme étant un cube, avec les coordonnées des sommets de la forme $(\pm 1, \pm 1, \pm 1)$ (Fig. 1). La relation projective de passage du repère affine associé au parallélépipède vers le plan image s'exprime sous la forme :

$$\begin{pmatrix} \alpha_1 u_1 & \dots & \alpha_8 u_8 \\ \alpha_1 v_1 & \dots & \alpha_8 v_8 \\ \alpha_1 & \dots & \alpha_8 \end{pmatrix} = \tilde{X} \cdot \begin{pmatrix} 1 & \dots & -1 \\ 1 & \dots & -1 \\ 1 & \dots & -1 \\ 1 & \dots & 1 \end{pmatrix}. \quad (2)$$

où \tilde{X} est la matrice de projection canonique de dimension 3×4 . Elle est égale à la matrice de projection M (voir section 2), à un changement de repère affine $\tilde{\Lambda}$ près :

$$\tilde{X} \sim M \cdot \tilde{\Lambda} \sim K \cdot [R \ t]. \quad (3)$$

La matrice $\tilde{\Lambda}$ dépend uniquement des paramètres intrinsèques du parallélépipède. En utilisant le paramétrage de la

1. Dans cette partie, nous appelons position (resp. l'orientation) la position (resp. l'orientation) *relative* du parallélépipède par rapport à la caméra.

Figure 1, cette matrice s'exprime de la manière suivante :

$$\tilde{\Lambda} = \begin{pmatrix} l_1 & l_2 c_{12} & l_3 c_{13} & 0 \\ 0 & l_2 s_{12} & l_3 \frac{c_{23} - c_{13} c_{12}}{s_{12}} & 0 \\ 0 & 0 & l_3 \sqrt{\frac{s_{12}^2 - c_{13}^2 s_{12}^2 - (c_{23} - c_{13} c_{12})^2}{s_{12}^2}} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (4)$$

où : $c_{ij} = \cos \theta_{ij}$, $s_{ij} = \sin \theta_{ij}$, $\theta_{ij} \in]0 \ \pi[$, $l_i > 0$. À noter que dans le cas d'un parallélépipède rectangle la matrice $\tilde{\Lambda}$ est alors diagonale.

Dans la suite nous utilisons la relation :

$$X \sim K \cdot R \cdot \Lambda, \quad (5)$$

où X et Λ sont les matrices 3×3 constituées des trois premières lignes et colonnes de \tilde{X} et $\tilde{\Lambda}$ respectivement.

La matrice de projection canonique d'un parallélogramme. De façon analogue, la transformation des coordonnées canoniques 2D d'un parallélogramme (soit d'un carré bi-unitaire) vers le plan image s'exprime de la manière suivante :

$$\begin{pmatrix} \alpha_1 u_1 & \dots & \alpha_4 u_4 \\ \alpha_1 v_1 & \dots & \alpha_4 v_4 \\ \alpha_1 & \dots & \alpha_4 \end{pmatrix} = \tilde{X} \cdot \begin{pmatrix} 1 & \dots & -1 \\ 1 & \dots & -1 \\ 1 & \dots & 1 \end{pmatrix}. \quad (6)$$

Ici, la matrice de projection canonique \tilde{X} est la matrice 3×3 :

$$\tilde{X} \sim H \cdot \tilde{\Lambda} \sim K \cdot [r_1 r_2 \ t]. \quad (7)$$

où H est l'homographie entre le plan du parallélogramme et le plan image, et r_1, r_2 sont les deux premières colonnes de la matrice de rotation entre le plan du parallélogramme et le plan image. La matrice $\tilde{\Lambda}$ représentant les paramètres intrinsèques d'un parallélogramme s'exprime de la manière suivante (en utilisant les notations introduites précédemment pour le parallélépipède) :

$$\tilde{\Lambda} = \begin{pmatrix} l_1 & l_2 c_{12} & 0 \\ 0 & l_2 s_{12} & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

En utilisant, ici dans le cas du parallélogramme, la notation Λ pour les deux premières colonnes et lignes de la matrice $\tilde{\Lambda}$, ainsi que la notation X pour les trois premières lignes et deux premières colonnes de la matrice \tilde{X} , nous obtenons une expression similaire à celle du parallélépipède (5) :

$$X \sim K \cdot [r_1 r_2] \cdot \Lambda. \quad (8)$$

3.2 Relation de dualité

La matrice X définie dans les paragraphes précédents, pour les parallélépipèdes et les parallélogrammes, contient toute l'information géométrique fournie par les projections de ces primitives. Soit $\omega = K^{-T} \cdot K^{-1}$ la matrice image de la conique absolue, cette matrice caractérisant les propriétés intrinsèques de la caméra. En utilisant l'équation (5)

(resp. (8) pour un parallélogramme) et en exploitant l'orthonormalité de la matrice de rotation R , nous obtenons la relation suivante :

$$X^T \cdot \omega \cdot X \sim \Lambda^T \cdot \Lambda \sim \mu. \quad (9)$$

L'expression ci-dessus illustre la dualité existant entre les caractéristiques intrinsèques d'une caméra et celles d'un parallélépipède (resp. d'un parallélogramme). Dans le cas d'un parallélépipède cette dualité est complète. En effet, la matrice X ne dépend que des projections des sommets du parallélépipède dans l'image. Une fois ces projections définies, les caractéristiques de la caméra et celles du parallélépipède sont mutuellement dépendantes. Fixer une caractéristique de l'un des ensembles intrinsèques engendre alors une contrainte sur l'autre ensemble de paramètres.

Pour un parallélogramme la dualité n'est pas complète. Un paramètre connu du parallélogramme donne en effet une contrainte sur les paramètres de la caméra, mais la réciproque n'est pas toujours vraie. Le tableau 1 récapitule les représentations intrinsèques des différentes entités.

L'entité	La représentation intrinsèque
caméra (w)	$\begin{pmatrix} \frac{1}{\alpha_u^2} & 0 & -\frac{u_0}{\alpha_u^2} \\ 0 & \frac{1}{\alpha_v^2} & -\frac{v_0}{\alpha_v^2} \\ -\frac{u_0}{\alpha_u^2} & -\frac{v_0}{\alpha_v^2} & 1 + \frac{u_0^2}{\alpha_u^2} + \frac{v_0^2}{\alpha_v^2} \end{pmatrix}$
parallélépipède (μ)	$\begin{pmatrix} l_1^2 & l_1 l_2 c_{12} & l_1 l_3 c_{13} \\ l_1 l_2 c_{12} & l_2^2 & l_2 l_3 c_{23} \\ l_1 l_3 c_{13} & l_2 l_3 c_{23} & l_3^2 \end{pmatrix}$
parallélogramme (μ)	$\begin{pmatrix} l_1^2 & l_1 l_2 c_{12} \\ l_1 l_2 c_{12} & l_2^2 \end{pmatrix}$

TAB. 1 – Les représentations intrinsèques de la caméra, du parallélépipède et du parallélogramme.

4 Calibrage

Dans cette section, nous décrivons comment utiliser les projections d'un parallélépipède pour calibrer une caméra à l'aide des connaissances sur le parallélépipède ou sur la caméra. L'étude est faite en détail pour les parallélépipèdes, celle des parallélogrammes pouvant en être déduite. Nous considérons, dans un premier temps, les connaissances *a priori* sur les paramètres intrinsèques de la caméra, ou du parallélépipède. Ces connaissances fournissent des contraintes qui sont exploitables pour une seule image et qui sont éventuellement suffisantes pour le calibrage de la caméra. Dans un deuxième temps, nous considérons les contraintes supplémentaires qui apparaissent lorsque plusieurs projections d'un même parallélépipède sont disponibles.

4.1 Connaissances a priori

Les connaissances a priori sur les paramètres intrinsèques de la caméra, respectivement du parallélépipède, donnent des contraintes sur l'entité duale, via la relation (9) introduite précédemment. Ces contraintes permettent de calibrer une caméra, souvent à partir d'une seule image, et en

même temps de reconstruire le parallélépipède dans l'espace. Cela constitue l'étape préliminaire d'une modélisation de la scène.

Les paragraphes suivants décrivent tout d'abord les différentes contraintes, puis introduisent les questions relatives au paramétrage du problème de calibrage. Dans la suite, la j^e colonne de la matrice X sera notée X_j .

Les connaissances a priori sur un parallélépipède ou un parallélogramme.

- Le fait que l'un des angles θ_{ij} du parallélépipède (ou du parallélogramme) soit un angle *droit*, conduit à la contrainte *linéaire* suivante sur ω , et donc sur les paramètres intrinsèques de la caméra :

$$X_i^T \cdot \omega \cdot X_j = 0 .$$

Ceci s'obtient en remplaçant $c_{ij} = \cos \theta_{ij} = 0$ dans la matrice μ concernée (voir le tableau 1) de l'équation (9) ;

- la connaissance d'un rapport de longueurs $r_{ij} = l_i/l_j$ conduit à la contrainte *linéaire* suivante :

$$(X_i^T \cdot \omega \cdot X_i) - r_{ij}^2 (X_j^T \cdot \omega \cdot X_j) = 0 ;$$

- la connaissance d'un angle θ_{ij} (non droit) conduit à la contrainte *quadratique* suivante :

$$(X_i^T \cdot \omega \cdot X_j)^2 - c_{ij}^2 (X_i^T \cdot \omega \cdot X_i)(X_j^T \cdot \omega \cdot X_j) = 0.$$

où $i, j \in \{1..3\}$ pour un parallélépipède et $i, j \in \{1,2\}$ pour un parallélogramme.

Les connaissances a priori sur la caméra. Dans le cas d'un parallélépipède, la dualité (9) peut aussi s'exprimer suivant :

$$\omega \sim Y^T \cdot \mu \cdot Y ,$$

où Y représente la matrice inverse de X .

Les connaissances sur les paramètres intrinsèques de la caméra donnent alors des contraintes sur μ :

- la connaissance des coordonnées image u_0 ou v_0 donne les contraintes *linéaires* suivantes :

$$(Y_1^T \cdot \mu \cdot Y_3) + u_0 (Y_1^T \cdot \mu \cdot Y_1) = 0 ;$$

$$(Y_2^T \cdot \mu \cdot Y_3) + v_0 (Y_2^T \cdot \mu \cdot Y_2) = 0 ;$$

- la connaissance du rapport d'échelle $\tau = \alpha_u/\alpha_v$ donne la contrainte *linéaire* suivante :

$$(Y_2^T \cdot \mu \cdot Y_2) - \tau^2 (Y_1^T \cdot \mu \cdot Y_1) = 0 ;$$

- quant à la distance focale, nous supposons ici que connaître sa valeur implique que tous les paramètres soient connus (ce qui est le cas en pratique), nous ne considérons donc pas le cas d'une connaissance isolée de la distance focale. Et si tous les paramètres intrinsèques de la caméra sont connus, μ peut être directement obtenue à l'aide de l'équation (9).

Le paramétrage. Il y a différentes manières de paramétrer le problème du calibrage et donc de prendre en compte les contraintes décrites ci-dessus. Ces contraintes peuvent être respectées exactement en utilisant un paramétrage minimal, ou bien constituer des équations supplémentaires dans une estimation aux moindres carrés.

La relation de dualité établie précédemment permet de considérer indifféremment les paramètres de μ ou ceux de ω comme inconnues du calibrage. Prenons par exemple le paramétrage du problème en μ . Les connaissances sur les paramètres intrinsèques de la caméra peuvent être prises en compte au travers des équations décrites ci-dessus, qui seront rajoutées au système d'équations linéaires (décrit plus loin au paragraphe §4.2). Quant aux connaissances sur le parallélépipède, il y a au moins deux manières de les exploiter :

1. la contrainte est utilisée pour réduire le nombre d'inconnues, par exemple la contrainte que l'angle θ_{12} est droit conduit au fait que la variable μ_{12} est dans ce cas égale à zéro ;
2. la contrainte donne une équation supplémentaire. Dans l'exemple, μ_{12} est gardée comme inconnue et l'équation $\mu_{12} = 0$ est rajoutée au système linéaire. Il est à noter ici qu'une résolution aux moindres carrés ne garantit pas que la valeur estimée de μ_{12} sera alors égale à zéro.

La deuxième manière permet donc de prendre en compte des connaissances approximatives telles quelles, idéalement à l'aide d'une résolution aux moindres carrés pondérés. Cette deuxième possibilité nécessite cependant une pondération adéquate des différents types de grandeurs (angles et longueurs). Notre implémentation est donc en pratique basée sur la première méthode.

Le paramétrage en ω est analogue à celui en μ . En pratique, nous paramétrons le problème en μ plutôt qu'en ω , ce qui est avantageux dans le cas de plusieurs images d'un seul parallélépipède (voir la section suivante).

En principe, il serait aussi possible de paramétrer en ω et μ à la fois. Dans ce cas, les connaissances sur ω seraient utilisées directement dans le paramétrage de ω , mais aussi comme équations sur μ , et réciproquement. En dehors des équations provenant de connaissances a priori, des équations linéaires exprimant l'égalité (à un facteur d'échelle près) de par exemple μ et $X^T \cdot \omega \cdot X$, devraient être rajoutées au système. Nous n'utilisons pas cette approche qui conduit à un système linéaire beaucoup plus grand et moins intuitif.

4.2 Contraintes multi-images

Lorsque plusieurs images d'un parallélépipède sont disponibles, il n'est plus nécessaire d'avoir des connaissances a priori pour calibrer. Dans le cas où un nombre suffisant d'images sont disponibles, des méthodes d'auto-calibrage peuvent être mises en œuvre [18]. Néanmoins, une méthode utilisant les structures parallèles possède un certain

nombre d'avantages :

- moins de correspondances sont nécessaires (cinq points et demi par image) ; les différentes images ne doivent pas forcément contenir les mêmes points, il faut juste pouvoir définir des parallélépipèdes communs ;
- les contraintes métriques supplémentaires sur un parallélépipède, telles que décrites précédemment, sont évidemment très utiles pour améliorer la stabilité et surtout pour réduire le nombre des paramètres à estimer ainsi que le nombre des singularités ;
- l'auto-calibrage générique doit déterminer (implicitement ou explicitement) la structure affine (la position du plan à l'infini) et puis la structure Euclidienne (la position de la conique absolue sur le plan à l'infini) de la scène, la partie affine étant la plus difficile. Dans notre cas, un parallélépipède nous fournit directement la structure affine (un cube, ou bien un carré dans le cas d'un parallélogramme). Ainsi, notre problème de calibrage est équivalent aux approches d'auto-calibrage utilisant des rotations pures [12, 6] ou des translations pures suivies de mouvements généraux [1, 19], et partage avec ces approches l'avantage d'une complexité algébrique inférieure à celle d'une approche d'auto-calibrage générique.

Dans la suite, nous considérons des contraintes de calibrage provenant de plusieurs images d'un parallélépipède (la prise en compte de plusieurs images de plusieurs parallélépipèdes constituant une de nos perspectives). Dans le reste du document, les exposants correspondent au numéro d'image, Y_1^i est par exemple la première colonne de la matrice Y associée à l'image i . Comme mentionné précédemment, nous paramétrons le problème en μ (les ω^i peuvent être calculées à partir de μ et des matrices X^i). Nous décrivons maintenant comment exploiter les contraintes disponibles.

Les connaissances a priori sur le parallélépipède sont directement utilisées afin de réduire le nombre des inconnues dans μ . Quant aux connaissances sur les paramètres intrinsèques de la caméra, nous distinguons trois cas :

- un paramètre *connu* fournit des équations *linéaires* (une par image) qui sont utilisées comme décrit au §4.1 ;
- un paramètre *inconnu* mais *constant* à travers les différentes images, fournit des équations sur μ qui sont *quadratiques* en général. Par exemple, la connaissance que u_0 est constant, donne l'équation suivante entre chaque paire d'images i et j :

$$\begin{aligned} & ((Y_1^i)^T \cdot \mu \cdot Y_3^i) \left((Y_1^j)^T \cdot \mu \cdot Y_1^j \right) \\ & - \left((Y_1^j)^T \cdot \mu \cdot Y_3^j \right) \left((Y_1^i)^T \cdot \mu \cdot Y_1^i \right) = 0. \end{aligned}$$

Il est à noter que si *tous* les paramètres intrinsèques de la caméra sont constants, des équations *linéaires* sont obtenues (voir le paragraphe suivant) ;

- un paramètre *inconnu* et *variable* (la distance focale si le zoom est utilisé) ne donne pas de contrainte.

Les paramètres intrinsèques de la caméra sont constants.

Comme cela a été montré par Hartley [12], les équations quadratiques mentionnées ci-dessus peuvent être modifiées en des équations linéaires : si toutes les X^i et leurs inverses Y^i sont normalisées (par multiplication avec des scalaires) afin d'avoir des déterminants égaux à 1, alors l'équation matricielle suivante est satisfaite *exactement* pour des paires d'images (et non à un facteur près comme en général) :

$$(Y^i)^T \cdot \mu \cdot Y^i = (Y^j)^T \cdot \mu \cdot Y^j.$$

Ces équations linéaires peuvent alors être utilisées conjointement à celles provenant de connaissances *a priori* sur quelques-uns des paramètres intrinsèques de la caméra.

4.3 Algorithme de calibrage

Notre approche consiste en deux étapes. D'abord, toutes les équations linéaires sont utilisées pour estimer μ (le système linéaire est résolu aux moindres carrés à l'aide d'une décomposition en valeurs singulières [10]). S'il y a une solution unique, le problème est résolu. Si par contre, le système linéaire est sous-contraint, alors les équations quadratiques provenant de paramètres de caméra *constants* peuvent réduire ou éliminer l'ambiguïté. Décider si le système est sous-contraint peut se faire sur la base d'une analyse de valeurs singulières du système. Celles-ci donne aussi le degré de l'ambiguïté (la dimension de l'espace des solutions). En pratique, nous avons toujours suffisamment de contraintes pour avoir une dimension de deux au maximum. Donc, deux équations quadratiques sont en général suffisantes pour obtenir un nombre fini de solutions. Si plus d'équations sont disponibles, la meilleure solution initiale peut être trouvée, par exemple, par une approche de type RANSAC [9].

5 Reconstruction

La méthode de calibrage présentée débouche naturellement sur une application simple et efficace de reconstruction de scène à partir d'une seule image ou de quelques images. Peu d'informations et d'interactions sont en effet suffisantes pour calibrer les caméras et construire un modèle tridimensionnel, l'utilisateur doit seulement cliquer quelques points dans les images pour définir les positions des primitives. Cette application se révèle être un outil rapide et intuitif pour construire des modèles à partir d'images d'origines quelconques, sur lesquelles aucune information n'est disponible.

La détermination de la pose. Lorsque plusieurs images sont utilisées pour la reconstruction, il est tout d'abord nécessaire d'exprimer les positions de toutes les caméras dans un même repère global. Cette étape est bien sûr inutile si une seule image est utilisée. Deux cas sont ici à considérer suivant le fait qu'une primitive soit présente dans toutes les

images ou non.

1. Toutes les images contiennent la projection d'une même primitive. Alors il suffit de choisir le repère de cette primitive comme référence. La matrice de projection canonique \bar{X}^i de cette primitive dans l'image i est : $\bar{X}^i = K \cdot [R^i t^i] \cdot \bar{\Lambda}$ (voir la partie 3.1). Comme les matrices K et $\bar{\Lambda}$ ont déjà été estimées par calibrage, la pose $[R^i t^i]$ de l'image i peut donc être calculée directement (en présence de bruit, la matrice R ne sera pas exactement une matrice de rotation, elle peut être ajustée en utilisant une décomposition SVD [15]).
2. Aucune primitive n'est visible dans toutes les images. Nous supposons alors que chaque image partage une primitive avec suffisamment d'autres images. On peut alors adapter la méthode décrite dans [20] et basée sur les parallélogrammes. Celle-ci permet d'estimer les poses de caméras à partir des homographies entre plusieurs plans (ici, les faces des parallélépipèdes ou d'autres parallélogrammes) et plusieurs images.

La reconstruction des primitives. Les primitives à reconstruire se départagent suivant deux catégories : les primitives utilisées pour le calibrage (les parallélépipèdes en général) et celles définies pour la reconstruction uniquement (des parallélogrammes ou des points par exemple). La reconstruction des primitives utilisées pour le calibrage est triviale : les paramètres intrinsèques des primitives sont en effet donnés par la relation de dualité (9). Dans le cas de plusieurs images, les algorithmes de détermination de pose décrits ci-dessus donnent les poses de ces primitives. Pour les primitives définies pour la reconstruction, on peut procéder de la manière suivante :

- reconstruire les points qui sont visibles dans plusieurs images en utilisant par exemple [11] ;
- reconstruire les points qui ne sont visibles que dans une seule image en utilisant les points déjà reconstruits dans l'étape précédente (pour obtenir une reconstruction cohérente) et des contraintes géométriques additionnelles. Ces contraintes telles que la colinéarité et la coplanarité (contraintes linéaires) sont nécessaires pour reconstruire des points visibles dans une image uniquement. La reconstruction de ces points est alors obtenue par résolution du système linéaire contenant l'ensemble des équations provenant des contraintes géométriques (pour plus de détails consulter [22]).

À noter que dans le cas d'une seule image, seule la deuxième étape est effectuée. L'intérêt majeur de cette approche est qu'un modèle peut être créé à partir d'images ne se recouvrant pas mais contenant une primitive commune (voir l'exemple de la Figure 5 dans la partie suivante).

L'algorithme. L'algorithme complet de calibrage et de reconstruction est le suivant :

1. Les projections d'un ou plusieurs parallélépipèdes et parallélogrammes sont définies dans les images et leurs

paramètres intrinsèques connus sont donnés. Ces paramètres sont utilisés pour calibrer les caméras.

2. Dans le cas de plusieurs images, les poses des caméras sont déterminées.
3. Les positions dans l'espace des points qui sont visibles dans plusieurs images sont calculées.
4. Les points additionnels du modèle et les contraintes linéaires qui leur sont associées sont définis (la coplanarité par exemple). Toutes les contraintes forment un système linéaire pour chaque image. Les points reconstruits à partir de plusieurs images sont inclus dans ce système pour obtenir une reconstruction cohérente avec le repère global.
5. Le modèle est créé à partir des points reconstruits et des images de départ (pour les textures). Ce modèle est constitué de facettes planes correspondant aux primitives définies par l'utilisateur.

Cet algorithme permet de calibrer et reconstruire par des méthodes linéaires. Les résultats obtenus peuvent naturellement être raffinés à l'aide d'une optimisation non-linéaire, cette optimisation prenant en compte toutes les informations données *a priori*. Les exemples présentés dans cet article ont été obtenus en utilisant des contraintes linéaires uniquement et sans post-optimisation.

6 Résultats expérimentaux

Une interface basée sur la librairie OpenGL a été développée à partir des idées décrites dans cet article. Cette interface permet, en quelques minutes, de construire des modèles réalistes à partir de photos non calibrées, et de les visualiser. Ces modèles construits peuvent être exportés au format standard VRML. Dans ce paragraphe, nous présentons quelques résultats illustratifs de reconstruction et de réalité augmentée.

Dans les exemples suivants, les caméras ont été calibrées à l'aide des angles droits d'un parallélépipède et en supposant que le point principal est au milieu de l'image comme mentionné et justifié dans [22].

La suppression d'éléments d'une scène. L'objectif ici est de simuler la suppression de l'escalier devant la maison, en utilisant l'image en haut de la Figure 3. Il ne s'agit pas seulement de le « gommer » dans cette image, mais de le supprimer dans le modèle virtuel *tridimensionnel* construit de la maison. Nous décrivons brièvement ici les étapes effectuées pour arriver au modèle virtuel corrigé. Deux images du modèle sont montrées en bas de la Figure 3.

La caméra a été calibrée à l'aide du parallélépipède montré partiellement en haut de la Figure 3 et de la connaissance des angles droits. Le parallélépipède a par la suite été reconstruit, puis des contraintes de coplanarité ont permis de reconstruire les autres points définissant le modèle (des points sur la façade et le sol). À ce stade nous disposons d'un modèle *géométrique* qui ne contient pas l'escalier. Afin d'obtenir des rendus réalistes, nous avons plaqué des textures sur les parties qui ne sont pas visibles dans

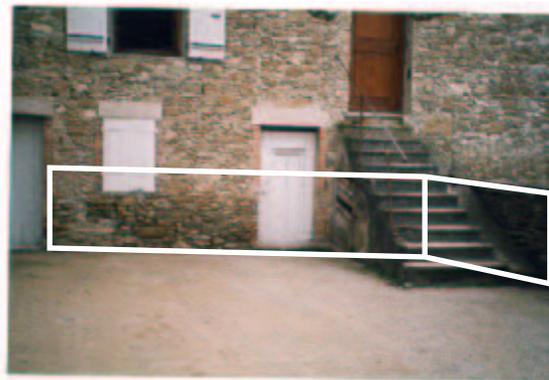


FIG. 3 – L'image originale de la maison (en haut) et deux images du modèle après la suppression de l'escalier.

l'image originale (cachées par l'escalier). Ces textures ont été prélevées dans d'autres parties du modèle, et ont été rectifiées perspectivement. Le calibrage et la reconstruction métrique de la scène ont, en effet, permis de corriger les effets perspectifs présents dans l'image originale.

La combinaison de modèles. Une deuxième application intéressante consiste à combiner différents modèles pour en créer de nouveaux. La Figure 4 montre une image d'une chambre et une autre d'un triptyque de la galerie Uffizi à Florence. Les primitives utilisées pour le calibrage sont surlignées en blanc. En bas de la Figure 4, des images d'une scène qui a été créée en combinant les modèles VRML, sont montrées.

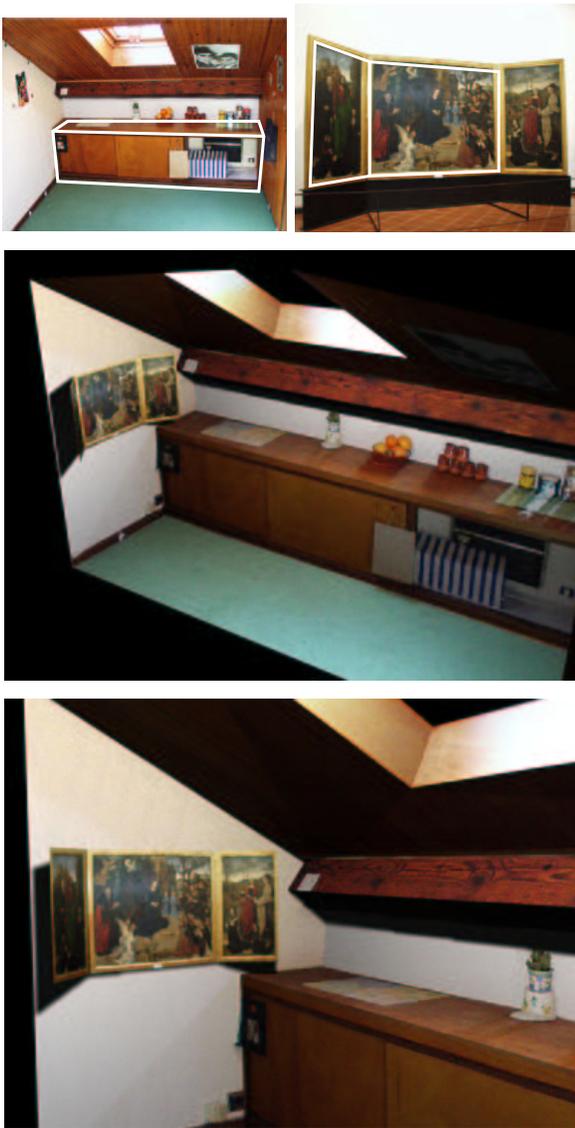


FIG. 4 – Les images originales d’une scène d’intérieur et d’un triptyque (en haut) et deux images d’une nouvelle scène combinée.

La reconstruction d’un bâtiment. L’approche de reconstruction décrite dans la section précédente permet de modéliser une scène à partir d’un petit nombre d’images. Nous montrons ici un exemple de reconstruction obtenue à partir de deux images (Figure 5). Ces deux images ont été prises selon deux points de vue diamétralement opposés du bâtiment. Le parallépipède utilisé pour le calibrage et la détermination des poses de la caméra est surligné en blanc dans les images. L’ensemble de la scène a été modélisé à partir de parallélogrammes et parallépipèdes définis par l’utilisateur (quatre parallépipèdes, quatre parallélogrammes ainsi que des contraintes de coplanarité pour reconstruire des points qui n’appartiennent à aucun parallélogramme).



FIG. 5 – Le bâtiment suivant deux points de vue opposés (en haut) et des images du modèle construit.

7 Conclusion

Nous avons présenté un ensemble d’outils mathématiques qui permettent le calibrage et l’acquisition de modèles tridimensionnels à partir d’images quelconques. Ces outils sont basés sur l’utilisation de contraintes géométriques au travers de primitives simples : les parallépipèdes et les parallélogrammes, ces primitives étant définies par l’utilisateur dans les images en quelques clics de souris.

Nous avons mis en évidence la dualité existant entre les ca-

ractéristiques intrinsèques de ces primitives et celles d'une caméra. Cette dualité permet de calibrer une caméra lorsque quelques informations sur la primitive sont connues ou lorsque quelques images d'une même scène sont disponibles. Les résultats expérimentaux présentés montrent l'intérêt de ces outils pour différents types d'applications de reconstruction et de réalité augmentée.

Nous travaillons actuellement sur l'amélioration de la méthode sur deux points : le premier consiste à élargir le domaine des contraintes exploitables pour le calibrage et la reconstruction. Nous avons mentionné par exemple le fait d'utiliser plusieurs parallélépipèdes dans plusieurs images pour calibrer ; un deuxième point concerne la réduction du nombre d'interactions nécessaire dans le cas de plusieurs images et donc l'automatisation partielle de la méthode.

Références

- [1] M. Armstrong, A. Zisserman, and P. Beardsley. Euclidean Structure from Uncalibrated Images. In *Proc. British Machine Vision Conf., York, England*, pp. 509–518, 1994.
- [2] B. Caprile and V. Torre. Using Vanishing Points for Camera Calibration. *International Journal of Computer Vision*, 4:127–140, 1990.
- [3] C.-S. Chen, C.-K. Yu, and Y.-P. Hung. New Calibration-free Approach for Augmented Reality Based on Parametrized Cuboid Structure. In *Proc. International Conf. on Computer Vision, Kerkyra, Greece*, pp. 30–37, 1999.
- [4] R. Cipolla and E. Boyer. 3D model acquisition from uncalibrated images. In *IAPR Workshop on Machine Vision Applications, Chiba, Japan*, pp. 559–568, 1998.
- [5] A. Criminisi, I. Reid, and A. Zisserman. Single View Metrology. In *Proc. International Conf. on Computer Vision, Kerkyra, Greece*, pp. 434–442, 1999.
- [6] L. de Agapito, R. Hartley, and E. Hayman. Linear selfcalibration of a rotating and zooming camera. In *Proc. Conf. on Computer Vision and Pattern Recognition, Fort Collins, USA*, pp. 15–21, 1999.
- [7] 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*, 1996.
- [8] O. Faugeras. Stratification of three-dimensional vision: Projective, affine and metric representations. *Journal of the Optical Society of America*, 12:465–484, 1995.
- [9] 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, 1981.
- [10] G.H. Golub and C.F. van Loan. *Matrix Computation*. The Johns Hopkins University Press, Baltimore, 1989.
- [11] R. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997.
- [12] R.I. Hartley. Self-calibration of Stationary Cameras. *International Journal of Computer Vision*, 22(1):5–23, 1997.
- [13] R.I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [14] Y. Horry, K.-I. Anjyo, and K. Arai. Tour Into the Picture: Using a Spidery Mesh Interface to Make Animation from a Single Image. *Computer Graphics*, 31:225–232, 1997.
- [15] K. Kanatani. *Statistical Optimization for Geometric Computation: Theory and Practice*. Elsevier Scienc, 1996.
- [16] D. Liebowitz and A. Zisserman. Metric Rectification for Perspective Images of Planes. In *Proc. Conf. on Computer Vision and Pattern Recognition, Santa Barbara, USA*, 1998.
- [17] D. Liebowitz and A. Zisserman. Combining Scene and Auto-calibration Constraints. In *Proc. International Conf. on Computer Vision, Kerkyra, Greece*, 1999.
- [18] 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.
- [19] M. Pollefeys, L. Van Gool, and M. Proesmans. Euclidean 3D Reconstruction from Image Sequences with Variable Focal Lengths. In *Proc. European Conf. on Computer Vision, Cambridge, England*, pp. 31–42, 1996.
- [20] P. Sturm. Algorithms for Plane-Based Pose Estimation. In *Proc. Conf. on Computer Vision and Pattern Recognition, Hilton Head Island, USA*, pp. 1010–1017, 2000.
- [21] P. Sturm and S. Maybank. On Plane-Based Camera Calibration: A General Algorithm, Singularities, Applications. In *Proc. Conf. on Computer Vision and Pattern Recognition, Fort Collins, USA*, pp. 432–437, 1999.
- [22] M. Wilczkowiak, E. Boyer, and P. Sturm. Camera Calibration and 3D Reconstruction from Single Images Using Parallelepipeds. In *Proc. International Conf. on Computer Vision, Vancouver, Canada*, pp. 142–148, 2001.
- [23] Z. Zhang. Flexible Camera Calibration By Viewing a Plane From Unknown Orientations. In *Proc. International Conf. on Computer Vision, Kerkyra, Greece*, 1999.