EXTENDED VERSION — SHORT VERSION APPEARED IN THE 9TH ICCV,
NICE, FRANCE, OCTOBER 2003.

Multiple-View Structure and Motion From Line Correspondences

Adrien Bartoli **Peter Sturm**
INRIA Rhône-Alpes, 655, avenue de l'Europe
38334 Saint Ismier cedex, France. *First.Last@inria.fr*

Abstract

We address the problem of camera motion and structure reconstruction from line correspondences across multiple views, from initialization to final bundle adjustment. One of the main difficulties when dealing with line features is their algebraic representation.

First, we consider the triangulation problem. Based on Plücker coordinates to represent the lines, we propose a maximum likelihood algorithm, relying on linearizing the Plücker constraint and on a Plücker correction procedure to compute the closest Plücker coordinates to a given 6-vector.

Second, we consider the bundle adjustment problem. Previous overparameterizations of 3D lines induce gauge freedoms and/or internal consistency constraints. We propose the orthonormal representation, which allows handy non-linear optimization of 3D lines using the minimum 4 parameters, within an unconstrained non-linear optimizer.

We compare our algorithms to existing ones on simulated and real data.

1. Introduction

The goal of this paper is to give methods for reconstruction of line features from image correspondences over multiple views, from initialization to final bundle adjustment. Reconstruction of line features is an important topic since it is used in areas such as scene modeling, augmented reality and visual servoing. Bundle adjustment is the computation of an optimal visual reconstruction of camera motion and 3D scene structure, where optimal means of maximum likelihood in terms of reprojected image error. We make no assumption about the calibration of the cameras. We assume that line correspondences over at least three views are available[1].

While the multiple-view geometry of lines is well-understood, see e.g. [3, 5], there is still a need for practical structure and motion algorithms. The factorization algorithms [7, 9, 13] yield reliable results but requires that all lines are visible in all views. We focus on the common three-stage approach, see e.g. [5, §17.5], consisting in (i) computing camera motion using inter-image matching tensors, (ii) triangulating the features and (iii) running bundle adjustment.

There exist reliable algorithms for step (i). In particular, it can be solved by computing trifocal tensors for triplets of consecutive images, using e.g. the automatic computation algorithm described in [5, §15.6], and registering the triplets in a manner similar to [4]. Other integrated motion estimation systems are [10], based on Kalman filtering techniques, and [14] based on registering each view in turn.

In steps (ii) and (iii), one of the main difficulties when dealing with line features arises: The algebraic representation. Indeed, there is no minimal, complete and globally non-singular parameterization of the 4-dimensional set of 3D lines, see e.g. [5, §2.2]. Hence, they are often overparameterized, e.g. as the join of two points or as the meet of two planes (8 parameters), or by the 6 coefficients of their Plücker coordinates, which must satisfy the bilinear Plücker constraint. Another overparameterization is two images of the line (6 parameters). The most appropriate representation depends upon the problem considered. For example, the algorithm in [5, §15.2] shows that the 'two image lines' representation is well-adapted to the computation of the trifocal tensor, while the sequential algorithm of [10] is based on Plücker coordinates.

Concerning step (ii), many of the previous works assume calibrated cameras, e.g. [6, 11, 12, 15] and use specific Euclidean representations. The linear three view algorithm of [15] and the algorithm of [12] utilize a 'closest point+direction' representation, while [11] uses the projections of the line on the $x = 0$ and the $y = 0$ planes, which has obvious singularities. These algorithms yield sub-optimal results in that none of them maximizes the individual likelihood of the reconstructed lines.

Bundle adjustment, step (iii), is a non-linear procedure involving camera and line parameters, which maximizes the

[1] line correspondences over two views do not constrain the camera motion.

likelihood of the reconstruction, corresponding to minimizing the reprojection error when the noise on measured features is assumed to have an identical and independent normal distribution. Previously-mentioned overparameterizations are not well-adapted to standard non-linear optimizers. The two point and the two plane overparameterizations have 4 degrees of internal gauge freedoms[2] which may induce numerical instabilities. The 'two image lines' parameterization has 2 degrees of internal gauge freedoms and implies that one may have to choose different images for different lines since all lines may not be visible in all views. Also, one must check that the chosen images are not too close to each other. Finally, direct optimization of Plücker coordinates makes sense only if a constrained optimization technique is used to enforce the bilinear Plücker constraint. An appropriate representation would not involve internal constraint or gauge freedom.

To summarize, there is a need for an efficient optimal triangulation algorithm, and a representation of 3D lines well-adapted to non-linear optimization. We address both of these problems through the following contributions.

In §3, we propose triangulation methods based on using Plücker coordinates to represent the lines. A simple and optimal algorithm is obtained based on linearizing the bilinear Plücker constraint within an iteratively reweighted least squares approach.

In §4, we propose a non-linear representation of 3D lines that we call the *orthonormal representation*. This representation allows efficient non-linear optimization since only the minimum 4 parameters are needed for optimization, within any unconstrained non-linear optimizer. With this representation, there is no internal gauge freedom or consistency constraint, and analytic differentiation of the error function is possible.

Finally, §5 validates our algorithms and compares them to existing ones. The next section gives some preliminaries and notations and formally states the problem.

2. Preliminaries and Notations

We make no formal distinction between coordinate vectors and physical entities. Everything is represented in homogeneous coordinates. Equality up to scale is denoted by $\sim$, transposition and transposed inverse by T and $^{-\mathsf{T}}$. Vectors are typeset using bold fonts ($\mathbf{L}$, $\mathbf{l}$), matrices using sans-serif fonts (S, A, R) and scalars in italics. Bars represent inhomogeneous leading parts of vectors or matrices, e.g. $\mathbf{M}^\mathsf{T} \sim \left(\bar{\mathbf{M}}^\mathsf{T} \mid m\right)$. The $\mathcal{L}_2$-norm of vector $\mathbf{v}$ is denoted $\|\mathbf{v}\|$. The identity matrix is denoted I. $SO(2)$ and $SO(3)$ denote the 2D and 3D rotation groups.

[2] for the former one, the position of the points along the line, and the free scale factor of the homogeneous representation of these points.

The 2D orthogonal (Euclidean) distance between point $\mathbf{q}$ and line $\mathbf{l}$ weighted by q_3 is:

$$d_\perp^2(\mathbf{q}, \mathbf{l}) = (\mathbf{q}^\mathsf{T}\mathbf{l})^2/(l_1^2 + l_2^2). \tag{1}$$

Plücker line coordinates. Given two 3D points $\mathbf{M}^\mathsf{T} \sim \left(\bar{\mathbf{M}}^\mathsf{T} \mid m\right)$ and $\mathbf{N}^\mathsf{T} \sim \left(\bar{\mathbf{N}}^\mathsf{T} \mid n\right)$, one can represent the line joining them by a homogeneous 'Plücker' 6-vector $\mathbf{L}^\mathsf{T} \sim \left(\mathbf{a}^\mathsf{T} \mid \mathbf{b}^\mathsf{T}\right)$, where $\mathbf{a} = \bar{\mathbf{M}} \times \bar{\mathbf{N}}$ and $\mathbf{b} = m\bar{\mathbf{N}} - n\bar{\mathbf{M}}$, see e.g. [5, §2.2]. Other conventions for Plücker 6-vectors are also possible. Each comes with a bilinear constraint that the 6-vector must satisfy in order to represent valid line coordinates. For our definition, the constraint is:

$$\mathcal{C}(\mathbf{L}) = 0 \text{ where } \mathcal{C}(\mathbf{L}) = \mathbf{a}^\mathsf{T}\mathbf{b}. \tag{2}$$

Perspective projection matrix for Plücker line coordinates. Given a standard (3×4) perspective projection matrix $\mathsf{P} \sim (\bar{\mathsf{P}} \mid \mathbf{p})$, a (3×6) matrix projecting Plücker line coordinates [2, 3] is given by:

$$\widetilde{\mathsf{P}} \sim (\det(\bar{\mathsf{P}})\bar{\mathsf{P}}^{-\mathsf{T}} \mid [\mathbf{p}]_\times \bar{\mathsf{P}}). \tag{3}$$

It can be easily derived by expanding the expression of the 2D line joining the projections of two points.

Maximum likelihood estimation. As noted in [5, §15.7.2], no matter how many points are used to represent an image line $\mathbf{l}$, the quadratic error function on it can be expressed in the form $d_\perp^2(\mathbf{x}, \mathbf{l}) + d_\perp^2(\mathbf{y}, \mathbf{l})$ for two weighted points $\mathbf{x}$, $\mathbf{y}$ on $\mathbf{l}$. We will use this representation for simplicity. If we have 3D lines $\mathcal{S} = \{\mathbf{L}^1, \ldots, \mathbf{L}^m\}$ and cameras $\mathcal{M} = \{\mathsf{P}^1, \ldots, \mathsf{P}^n\}$, the negative log likelihood function $\mathcal{E}(\mathcal{S}, \mathcal{M})$ for the reconstruction, corresponding to the reprojection error, can be written in terms of individual reprojection errors $\mathcal{E}(\mathbf{L}^j, \mathcal{M})$ for each line j:

$$\mathcal{E}(\mathcal{S}, \mathcal{M}) = \sum_{j=1}^{m} \mathcal{E}(\mathbf{L}^j, \mathcal{M}) \tag{4}$$

$$\mathcal{E}(\mathbf{L}^j, \mathcal{M}) = \sum_{i=1}^{n} \left(d_\perp^2(\mathbf{x}^{ij}, \mathbf{l}^{ij}) + d_\perp^2(\mathbf{y}^{ij}, \mathbf{l}^{ij})\right). \tag{5}$$

3. Triangulation

This section discusses computation of structure given camera motion. We propose direct linear and iterative non-linear methods to recover Plücker line coordinates. These algorithms are general in the sense that they can be used with calibrated, partially calibrated or uncalibrated cameras.

First, we describe a somehow trivial linear algorithm where a biased error function (compared to the reprojection

error) is minimized. This algorithm is subject to the same kind of drawback as the 8 point algorithm for computing the fundamental matrix: due to possible noise in the data, the resulting 6-vectors do not generally satisfy the bilinear Plücker constraint (2), similarly to the matrix computed by the 8 point algorithm not being rank deficient [5, §10.2]. We propose what we call a *Plücker correction* procedure, which allows to compute the closest Plücker coordinates to a 6-vector.

Second, we propose an algorithm where the reprojection error of the line is minimized. The cornerstone of this algorithm is the linearization of the Plücker constraint.

Since the reconstruction of each line is independent from the others, we drop the j index in this section.

3.1. Linear Algorithm

We describe a linear algorithm, 'LIN'. In the reprojection error (5), each term is based on the square of the 2D point-to-line orthogonal distance $d_\perp$, defined by equation (1). The denominator of this distance is the cause of the non-linearity. Ignoring this denominator leads to an algebraic distance denoted by d_a, biased compared to the orthogonal distance. d_a is linear in the predicted line $\mathbf{l}$ and defined by $d_a^2(\mathbf{q},\mathbf{l}) = d_\perp^2(\mathbf{q},\mathbf{l})\ w^2 = (\mathbf{q}^\mathsf{T}\mathbf{l})^2$, where the scalar factor w encapsulates the bias as $w^2 = l_1^2 + l_2^2$:

$$\left(w^i\right)^2 = \left((\widetilde{\mathsf{P}}^i\mathbf{L})_1\right)^2 + \left((\widetilde{\mathsf{P}}^i\mathbf{L})_2\right)^2. \tag{6}$$

We define the biased linear least squares error function:

$$\mathcal{B}(\mathbf{L},\mathcal{M}) = \sum_{i=1}^{n}\left((\mathbf{x}^{i\mathsf{T}}\widetilde{\mathsf{P}}^i\mathbf{L})^2 + (\mathbf{y}^{i\mathsf{T}}\widetilde{\mathsf{P}}^i\mathbf{L})^2\right) \tag{7}$$

$$= \|\mathsf{A}_{(2n\times 6)}\mathbf{L}\|^2,\ \mathsf{A} = \begin{pmatrix} \dots \\ \mathbf{x}^{i\mathsf{T}}\widetilde{\mathsf{P}}^i \\ \mathbf{y}^{i\mathsf{T}}\widetilde{\mathsf{P}}^i \\ \dots \end{pmatrix}. \tag{8}$$

Since $\mathbf{L}$ is an homogeneous vector, we add the constraint $\|\mathbf{L}\|^2 = 1$. The $\mathbf{L}$ that minimizes $\mathcal{B}(\mathbf{L},\mathcal{M})$ is then given by the singular vector of A associated to its smallest singular value. Due to noise, the recovered 6-vector does not in general satisfy the Plücker constraint (2).

3.2. Plücker Correction

Let $\mathbf{L}^\mathsf{T} \sim (\mathbf{a}^\mathsf{T} \mid \mathbf{b}^\mathsf{T})$ be a 6-vector that does not necessarily satisfy the Plücker constraint (2), i.e. $\mathbf{a}^\mathsf{T}\mathbf{b}$ might be non-zero. We seek $\widehat{\mathbf{L}}^\mathsf{T} \sim (\mathbf{u}^\mathsf{T} \mid \mathbf{v}^\mathsf{T})$, defined by $\min_{\widehat{\mathbf{L}},\mathbf{u}^\mathsf{T}\mathbf{v}=0}\|\widehat{\mathbf{L}} - \mathbf{L}\|^2$. Although this problem has a clear and concise formulation, it is *not* trivial.

We propose the following solution, summarized in a practical manner in table 1. We transform the original 3D problem to an equivalent 2D problem, solve the 2D transformed problem and bring the solution back to the original 3D coordinate frame.

A geometric interpretation. We interpret the 3-vectors $\mathbf{a}$, $\mathbf{b}$, $\mathbf{u}$ and $\mathbf{v}$ as coordinates of 3D points. These points are not directly linked to the underlying 3D line. This interpretation is just intended to visualize the problem. The Plücker constraint $\mathbf{u}^\mathsf{T}\mathbf{v}$ corresponds to the fact that the lines induced by the origin with $\mathbf{u}$ and $\mathbf{v}$ are perpendicular. The correction criterion is the sum of squared Euclidean distances between $\mathbf{a}$ and $\mathbf{u}$ and between $\mathbf{b}$ and $\mathbf{v}$. Hence, the problem may be formulated as finding two points $\mathbf{u}$ and $\mathbf{v}$, the closest as possible to $\mathbf{a}$ and $\mathbf{b}$ respectively and such that the lines induced by the origin with $\mathbf{u}$ and $\mathbf{v}$ are perpendicular. We begin by rotating the coordinate frame such that $\mathbf{a}$ and $\mathbf{b}$ are transferred on the $z = 0$ plane. This is the *reduction of the problem*. We *solve the reduced problem*, by finding two points on the $z = 0$ plane, minimizing the correction criterion and satisfying the Plücker constraint. Finally, we *express the solution* back to the original space.

Reducing the problem. Let us define the (3×2) matrices $\bar{\mathsf{C}} \sim (\mathbf{a}\ \mathbf{b})$ and $\widehat{\mathsf{C}} \sim (\mathbf{u}\ \mathbf{v})$. The Plücker constraint is fulfilled if and only if the columns of matrix $\widehat{\mathsf{C}}$ are orthogonal. We rewrite the correction criterion as :

$$\mathcal{O} = \|\mathbf{L} - \widehat{\mathbf{L}}\|^2 = \|\bar{\mathsf{C}} - \widehat{\mathsf{C}}\|^2.$$

Using the following singular value decomposition $\bar{\mathsf{C}}_{(3\times 2)} = \bar{\mathsf{U}}_{(3\times 2)}\bar{\mathsf{\Sigma}}_{(2\times 2)}\bar{\mathsf{V}}^\mathsf{T}_{(2\times 2)}$:

$$\mathcal{O} = \|\bar{\mathsf{U}}\bar{\mathsf{\Sigma}}\bar{\mathsf{V}}^\mathsf{T} - \widehat{\mathsf{C}}\|^2 = \|\bar{\mathsf{\Sigma}}\bar{\mathsf{V}}^\mathsf{T} - \bar{\mathsf{U}}^\mathsf{T}\widehat{\mathsf{C}}\|^2,$$

since $\bar{\mathsf{U}}$ has orthonormal columns. We define $\bar{\mathsf{Z}} = \bar{\mathsf{\Sigma}}\bar{\mathsf{V}}^\mathsf{T}$ and $\widehat{\mathsf{Z}} = \bar{\mathsf{U}}^\mathsf{T}\widehat{\mathsf{C}}$. Matrix $\bar{\mathsf{V}}$ is orthonormal and $\bar{\mathsf{\Sigma}}$ is diagonal, hence the rows of $\bar{\mathsf{Z}}$ are orthogonal (i.e. $\bar{\mathsf{Z}}\bar{\mathsf{Z}}^\mathsf{T}$ is diagonal, but not $\bar{\mathsf{Z}}^\mathsf{T}\bar{\mathsf{Z}}$). Note that $\widehat{\mathsf{Z}} = \bar{\mathsf{U}}^\mathsf{T}\widehat{\mathsf{C}}$ implies $\widehat{\mathsf{C}} = \bar{\mathsf{U}}\widehat{\mathsf{Z}}$, even if $\bar{\mathsf{U}}\bar{\mathsf{U}}^\mathsf{T}$ is not the identity[3]. The problem is reduced to finding a column-orthogonal[4] matrix $\widehat{\mathsf{Z}}$, as close as possible to the row-orthogonal matrix $\bar{\mathsf{Z}}$.

[3]Indeed, denote $\mathbf{u}_i$ the columns of matrix $\bar{\mathsf{U}}$ and form $\mathsf{U} = (\mathbf{u}_1\ \mathbf{u}_2\ \mathbf{u}_1\times\mathbf{u}_2)$. We have $\mathsf{U}^\mathsf{T}\bar{\mathsf{U}} = (\ \mathrm{I}_{(2\times 2)}\ \ \mathbf{0}_{(2\times 1)})^\mathsf{T}$. Let us multiply the correction criterion by U^T : $\mathcal{O} = \|(\ \bar{\mathsf{V}}\bar{\mathsf{\Sigma}}\ \ \mathbf{0}_{(2\times 1)})^\mathsf{T} - \mathsf{U}^\mathsf{T}\widehat{\mathsf{C}}\|^2$. Denote $\mathsf{Y}_{(3\times 2)} = \mathsf{U}^\mathsf{T}\widehat{\mathsf{C}}$. The optimal solution has the form $\mathsf{Y}^\mathsf{T} = (\ \widehat{\mathsf{Z}}^\mathsf{T}\ \ \mathbf{0}_{(2\times 1)})$, since, according to the geometric interpretation, the corrected points $\mathbf{u}$ and $\mathbf{v}$ must lie on the plane defined by points $\mathbf{a}$, $\mathbf{b}$ and the origin, the plane $z = 0$. Therefore, we obtain $\widehat{\mathsf{C}} = \mathsf{U}\mathsf{Y} = \bar{\mathsf{U}}\bar{\mathsf{Y}}$.

[4]The fact that matrix $\widehat{\mathsf{Z}} = \bar{\mathsf{U}}^\mathsf{T}\widehat{\mathsf{C}}$ is column-orthogonal is induced from the Plücker constraint. Indeed, this constraint implies that $\widehat{\mathsf{C}}$ is column-orthogonal, hence $\widehat{\mathsf{C}}^\mathsf{T}\widehat{\mathsf{C}}$ is diagonal. Matrix $\mathsf{U}^\mathsf{T}\widehat{\mathsf{C}}$, where $SO(3) \ni \mathsf{U} = (\mathbf{u}_1\ \mathbf{u}_2\ \mathbf{u}_1\times\mathbf{u}_2) = (\bar{\mathsf{U}}\ \bar{\mathbf{u}})$, is also column-orthogonal. Observe that $\widehat{\mathsf{C}}^\mathsf{T}\mathsf{U}\mathsf{U}^\mathsf{T}\widehat{\mathsf{C}} = \widehat{\mathsf{C}}^\mathsf{T}\bar{\mathsf{U}}\bar{\mathsf{U}}^\mathsf{T}\widehat{\mathsf{C}} + \widehat{\mathsf{C}}^\mathsf{T}\bar{\mathbf{u}}\bar{\mathbf{u}}^\mathsf{T}\widehat{\mathsf{C}} = \widehat{\mathsf{C}}^\mathsf{T}\bar{\mathsf{U}}\bar{\mathsf{U}}^\mathsf{T}\widehat{\mathsf{C}}$ since $\bar{\mathbf{u}}^\mathsf{T}\widehat{\mathsf{C}} = \mathbf{0}^\mathsf{T}$. Hence, matrix $\bar{\mathsf{U}}^\mathsf{T}\widehat{\mathsf{C}}$ is column-orthogonal.

Solving the reduced problem. We parameterize the column-orthogonal matrix $\widehat{\mathsf{Z}}$ as $\widehat{\mathsf{Z}} = \widehat{\mathsf{V}}\widehat{\Sigma}$, where $\widehat{\mathsf{V}}$ is orthonormal and $\widehat{\Sigma}$ is diagonal. Hence :

$$\mathcal{O} = \|\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T} - \widehat{\mathsf{V}}\widehat{\Sigma}\|^2 = \|\widehat{\mathsf{V}}^\mathsf{T}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T} - \widehat{\Sigma}\|^2.$$

The diagonal matrix $\widehat{\Sigma}$ which minimizes this expression is given by the diagonal entries of $\widehat{\mathsf{V}}^\mathsf{T}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T}$, and does not depend on the solution for $\widehat{\mathsf{V}}$. The orthonormal matrix $\widehat{\mathsf{V}} = (\widehat{\mathbf{v}}_1\ \widehat{\mathbf{v}}_2)$ is given by minimizing the sum of squares of the off-diagonal entries of $\widehat{\mathsf{V}}^\mathsf{T}\bar{\mathsf{Z}}$, with $\bar{\mathsf{Z}} = \bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T} = (\mathbf{z}_1\ \mathbf{z}_2)$:

$$\mathcal{O} = (\widehat{\mathbf{v}}_1^\mathsf{T}\mathbf{z}_2)^2 + (\widehat{\mathbf{v}}_2^\mathsf{T}\mathbf{z}_1)^2.$$

Define the rotation matrix with angle $\pi/2$ $\mathsf{M} = \left(\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right)$ and parameterize the orthonormal matrix $\widehat{\mathsf{V}}$ by a unit vector $\widehat{\mathbf{v}}$, as :

$$\left\{ \begin{array}{rcl} \widehat{\mathbf{v}}_1 & = & \widehat{\mathbf{v}} \\ \widehat{\mathbf{v}}_2 & = & \mathsf{M}\widehat{\mathbf{v}}, \end{array} \right.$$

The correction criterion can be rewritten as :

$$\mathcal{O} = (\widehat{\mathbf{v}}^\mathsf{T}\mathbf{z}_2)^2 + (\widehat{\mathbf{v}}^\mathsf{T}\mathsf{M}^\mathsf{T}\mathbf{z}_1)^2 = \|\mathsf{T}\widehat{\mathbf{v}}\|^2 \text{ avec } \mathsf{T} = \begin{pmatrix} \mathbf{z}_2^\mathsf{T} \\ \mathbf{z}_1^\mathsf{T}\mathsf{M} \end{pmatrix}.$$

The unit vector $\widehat{\mathbf{v}}$ minimizing this expression is given by the singular vector associated to the smallest singular value of matrix T.

Expressing the solution. From vector $\widehat{\mathbf{v}}$ which solves the reduced problem, we form the orthonormal matrix $\widehat{\mathsf{V}} = \left(\begin{smallmatrix} \hat{v}_1 & -\hat{v}_2 \\ \hat{v}_2 & \hat{v}_1 \end{smallmatrix}\right)$. The diagonal matrix $\widehat{\Sigma}$ is given by $\widehat{\Sigma} = \mathrm{diag}(\widehat{\mathsf{V}}^\mathsf{T}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T})$. Table 1 summarizes this algorithm.

- Compute the singular value decomposition $(\mathbf{a}\ \mathbf{b}) = \bar{\mathsf{U}}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T}$.
- Let $\bar{\mathsf{Z}} = \bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T}$, form matrix $\mathsf{T} = \left(\begin{smallmatrix} z_{21} & z_{22} \\ z_{12} & -z_{11} \end{smallmatrix}\right)$.
- Compute the singular vector $\widehat{\mathbf{v}}$ associated to the smallest singular value of matrix T.
- Let $\bar{\mathsf{V}} = \left(\begin{smallmatrix} \hat{v}_1 & -\hat{v}_2 \\ \hat{v}_2 & \hat{v}_1 \end{smallmatrix}\right)$, we obtain: $(\mathbf{u}\ \mathbf{v}) \sim \bar{\mathsf{U}}\ \widehat{\mathsf{V}}\ \mathrm{diag}\left(\widehat{\mathsf{V}}^\mathsf{T}\bar{\Sigma}\bar{\mathsf{V}}^\mathsf{T}\right)$.

Table 1. The *Plücker correction* algorithm. Given a 6-vector $\mathbf{L}^\mathsf{T} \sim (\mathbf{a}^\mathsf{T} \mid \mathbf{b}^\mathsf{T})$, this algorithm computes the closest Plücker coordinates $\widehat{\mathbf{L}}^\mathsf{T} \sim (\mathbf{u}^\mathsf{T} \mid \mathbf{v}^\mathsf{T})$, i.e. $\mathbf{u}^\mathsf{T}\mathbf{v} = 0$, in the sense of the $\mathcal{L}_2$-norm, i.e. $\|\widehat{\mathbf{L}} - \mathbf{L}\|^2$ is minimized.

3.3. Quasi-Linear Algorithms

We describe algorithms 'QLIN1' and 'QLIN2', that consider the reprojection error (5). They are based on an iterative bias-correction, through reweighting of the biased error function (7). Such algorithms are coined quasi-linear.

We showed previously that the orthogonal and the algebraic distances are related by a scalar factor, given by equation (6), depending on the 3D line. The reprojection error and the biased error functions are therefore related by a set of such factors, one for each image of the line. The fact that these factors depend on the unknown 3D line suggests an iterative reweighting scheme.

The first approach that comes to mind is 'QLIN1'. The linear system considered for method LIN is formed and solved. The resulting 6-vector $\mathbf{L}_0$ is corrected to be valid Plücker coordinates. This yields a biased estimate of the 3D line. Using this estimate, weight factors that contain the bias of the linear least squares error function are computed, and used to reweight the equations. The process is iterated to compute successive refined estimates $\mathbf{L}_k$ until convergence, where k is the iteration counter. Convergence is determined by thresholding the difference between two consecutive errors. It is typically reached in 3 or 4 iterations.

Experimental results show that this naive approach performs very badly, see §5. This is due to the fact that the Plücker constraint is enforced afterhand and is not taken into account while solving the linear least squares system.

To remedy to this problem, we propose 'QLIN2', that linearizes and enforces the Plücker constraint (2), as follows. The algorithm is summarized in table 2. Rewrite the constraint as $\mathcal{C}(\mathbf{L}) = \mathbf{L}^\mathsf{T}\mathsf{G}\mathbf{L}$ where $\mathsf{G}_{(6\times6)} = \left(\begin{smallmatrix} 0 & \mathrm{I} \\ \mathrm{I} & 0 \end{smallmatrix}\right)$. By expanding this expression to first order around the estimate $\mathbf{L}_k$, and after some minor algebraic manipulations, we obtain the following linear constraint on $\mathbf{L}_{k+1}$:

$$\mathcal{C}_k(\mathbf{L}_{k+1}) = \mathbf{L}_k^\mathsf{T}\mathsf{G}\mathbf{L}_{k+1} = 0.$$

We follow the constrained linear least squares optimization method summarized in [5, §A3.4.3] to enforce this linearized constraint, as well as $\|\mathbf{L}_{k+1}\| = 1$. The idea is to find an orthonormal basis of all possible vectors satisfying the constraint and to solve for a 5-vector $\boldsymbol{\gamma}$ expressed in this basis. Such an orthonormal basis is provided by computing the nullspace of $\mathbf{L}_k^\mathsf{T}\mathsf{G}$ using SVD. Let $\bar{\mathsf{V}}$ be a (6×5) orthonormal matrix whose columns span the basis (i.e. $\mathbf{L}_k^\mathsf{T}\mathsf{G}\bar{\mathsf{V}} = \mathbf{0}$), we define $\mathbf{L}_{k+1} = \bar{\mathsf{V}}\boldsymbol{\gamma}$, hence $\mathcal{C}_k(\mathbf{L}_{k+1}) = \mathbf{L}_k^\mathsf{T}\mathsf{G}\bar{\mathsf{V}}\boldsymbol{\gamma} = 0$ and $\|\mathbf{L}_{k+1}\| = \|\boldsymbol{\gamma}\|$. We solve for $\boldsymbol{\gamma}$ be substituting in equation (8) ($\|\mathsf{A}\mathbf{L}_{k+1}\|^2 = \|\mathsf{A}\bar{\mathsf{V}}\boldsymbol{\gamma}\|^2$). The singular vector associated to the smallest singular value of matrix $\mathsf{A}\bar{\mathsf{V}}$ provides the solution vector with unit norm-two such that $\mathcal{B}(\mathbf{L}_{k+1}, \mathcal{M})$ is minimized.

1. *Initialization:* Form the linear least squares system A from equation (8), compute $\mathbf{L}_0$ by minimizing $\|\mathsf{A}\mathbf{L}_0\|^2$, see §3.1, and by applying the Plücker correction procedure described in §3.2. Set $k = 0$.
2. *Constraint linearization:* Compute the SVD $\mathbf{L}_k^\mathsf{T}\mathsf{G} \sim \mathbf{u}^\mathsf{T}\mathrm{diag}(1,0,0,0,0,0)(\mathbf{v}_{(6\times1)} \mid \bar{\mathsf{V}}_{(6\times5)})^\mathsf{T}$.
3. *Estimation:* Compute $\min_{\boldsymbol{\gamma},\|\boldsymbol{\gamma}\|^2=1} \|\mathsf{A}\bar{\mathsf{V}}\boldsymbol{\gamma}\|^2$ and set $\mathbf{L}_{k+1} = \bar{\mathsf{V}}\boldsymbol{\gamma}$.
4. *Bias-correction:* Reweight the linear system A by computing the weights according to equation (6).
5. *Iteration:* Iterate steps 2, 3 and 4 until convergence.

Table 2. Quasi-linear algorithm 'QLIN2' for optimal triangulation.

4. Bundle Adjustment

Bundle adjustment is the non-linear minimization of the reprojection error (4), over camera and line parameters. We focus on the parameterization of 3D lines. Parameterizing the camera motion has been addressed in e.g. [1, 5, §A4.6].

4.1. Problem Statement

As said in the introduction, there are various possibilities to overparameterize the 4-dimensional set of 3D lines. In the context of non-linear optimization, choosing an overparameterized representation may induce the following problems. First, the computational cost of each iteration is increased by superfluous parameters. Second, artificial freedoms in the parameter set (internal gauge freedoms) are induced and may give rise to numerical instabilities. Third, some internal consistency constraints, such as the Plücker constraint, may have to be enforced.

These reasons motivate the need for a representation of 3D lines allowing non-linear optimization with the minimum 4 parameters. In that case, there is no free scale induced by homogeneity or internal consistency constraints, and any unconstrained non-linear optimizer can be used.

4.2. The Orthonormal Representation

The orthonormal representation has been introduced in [1] for the non-linear optimization of the fundamental matrix with the minimum 7 parameters. It consists in finding a representation involving elements of $SO(n)$ and scalars (hence the term 'orthonormal representation'). In particular, no other algebraic constraints should be necessary, such as the rank-two constraint of fundamental matrices or the bilinear Plücker constraint. Using orthonormal matrices implies that the representation is well-conditioned. Based on such a representation, local update using the minimum number of parameters is possible.

Commonly used non-linear optimizers, e.g. Newton-type optimizers such as Levenberg-Marquardt, often require the Jacobian matrix of the error function with respect to the update parameters. In the orthonormal representation framework, we split it as the product of the Jacobian matrix of the error function considered with respect to the 'standard' entity representation, e.g. the fundamental matrix or Plücker coordinates, and the *orthonormal Jacobian matrix*, i.e. for the 'standard' representation with respect to the update parameters.

Example: representing $\mathbb{P}^1$. We derive the orthonormal representation of the 1-dimensional projective space $\mathbb{P}^1$. This is used in §4.3 to derive the orthonormal representation of 3D lines. Let $\boldsymbol{\sigma} \in \mathbb{P}^1$. Such a 2-vector is defined up to scale and has therefore only 1 degree of freedom. We represent it by an $SO(2)$ matrix W defined by:

$$\mathsf{W} = \frac{1}{\|\boldsymbol{\sigma}\|}\begin{pmatrix}\sigma_1 & -\sigma_2 \\ \sigma_2 & \sigma_1\end{pmatrix}. \tag{9}$$

The first column of this matrix is $\boldsymbol{\sigma}$ itself, normalized to unit-norm. Let θ be the update parameter. A local update step is $\mathsf{W} \leftarrow \mathsf{W}\mathsf{R}(\theta)$ where $\mathsf{R}(\theta)$ is the 2D rotation matrix of angle θ. The Jacobian matrix $\frac{\partial\boldsymbol{\sigma}}{\partial\theta}$ evaluated at $\theta_0 = 0$ (the update is with respect to a base rotation) is given by:

$$\left.\frac{\partial\boldsymbol{\sigma}}{\partial\theta}\right|_{\theta_0} = \left.\frac{\partial\mathbf{w}_1}{\partial\theta}\right|_{\theta_0} = \begin{pmatrix}-\sigma_2 \\ \sigma_1\end{pmatrix} = \mathbf{w}_2, \tag{10}$$

where $\mathbf{w}_i$ is the i-th column of W.

Updating $SO(3)$. A matrix $\mathsf{U} \in SO(3)$ can be locally updated using 3 parameters by any well-behaved (locally non-singular) representation, such as 3 Euler angles $\boldsymbol{\theta}^\mathsf{T} = (\theta_1 \mid \theta_2 \mid \theta_3)$ as:

$$\mathsf{U} \leftarrow \mathsf{U}\mathsf{R}(\boldsymbol{\theta}) \text{ with } \mathsf{R}(\boldsymbol{\theta}) = \mathsf{R}_\mathrm{x}(\theta_1)\mathsf{R}_\mathrm{y}(\theta_2)\mathsf{R}_\mathrm{z}(\theta_3), \tag{11}$$

where $\mathsf{R}_\mathrm{x}(\theta_1)$, $\mathsf{R}_\mathrm{y}(\theta_2)$ and $\mathsf{R}_\mathrm{z}(\theta_3)$ are $SO(3)$ matrices representating 3D rotations around the x-, y- and z-axis of angle θ_1, θ_2 and θ_3 respectively. The Jacobian matrix is derived as follows. As in the $SO(2)$ case, the update is with respect to a base rotation. The orthonormal Jacobian matrix is therefore evaluated at $\boldsymbol{\theta}_0 = \mathbf{0}_{(3\times1)}$:

$$\left.\frac{\partial\mathsf{U}}{\partial\boldsymbol{\theta}}\right|_{\boldsymbol{\theta}_0} = \left(\left.\frac{\partial\mathsf{U}}{\partial\theta_1}\right|_{\boldsymbol{\theta}_0} \mid \left.\frac{\partial\mathsf{U}}{\partial\theta_2}\right|_{\boldsymbol{\theta}_0} \mid \left.\frac{\partial\mathsf{U}}{\partial\theta_3}\right|_{\boldsymbol{\theta}_0}\right).$$

After minor algebraic manipulations, we obtain:

$$\begin{aligned}\left.\frac{\partial\mathsf{U}}{\partial\theta_1}\right|_{\boldsymbol{\theta}_0} &= \left.\frac{\partial\left(\mathsf{U}\mathsf{R}_\mathrm{x}(\theta_1)\mathsf{R}_\mathrm{y}(\theta_2)\mathsf{R}_\mathrm{z}(\theta_3)\right)}{\partial\theta_1}\right|_{\boldsymbol{\theta}_0} \\ &= (\mathbf{0}_3 \mid \mathbf{u}_3 \mid -\mathbf{u}_2),\end{aligned} \tag{12}$$

where $\mathbf{u}_i$ is the i-th column of U. Similarly:

$$\left.\frac{\partial \mathsf{U}}{\partial \theta_2}\right|_{\boldsymbol{\theta}_0} = (-\mathbf{u}_3 \mid \mathbf{0}_3 \mid \mathbf{u}_1) \quad (13)$$

$$\left.\frac{\partial \mathsf{U}}{\partial \theta_3}\right|_{\boldsymbol{\theta}_0} = (\mathbf{u}_2 \mid -\mathbf{u}_1 \mid \mathbf{0}_3)\,. \quad (14)$$

These expressions are vectorized to form the orthonormal Jacobian matrix.

4.3. The Case of 3D Lines

The case of 3D lines is strongly linked with the cases of $SO(2)$ and $SO(3)$, as shown by the following result:

Any (projective) 3D line $\mathbf{L}$ *can be represented by:*

$$(\mathsf{U}, \mathsf{W}) \in SO(3) \times SO(2),$$

where $SO(2)$ *and* $SO(3)$ *are the Lie groups of respectively* (2×2) *and* (3×3) *rotation matrices.* (U, W) *is the orthonormal representation of the 3D line* $\mathbf{L}$*.*

The proof of this result is obtained by showing that any 3D line has an orthonormal representation $(\mathsf{U}, \mathsf{W}) \in SO(3) \times SO(2)$, while any $(\mathsf{U}, \mathsf{W}) \in SO(3) \times SO(2)$ corresponds to a unique 3D line. The next paragraph illustrates this by means of Plücker coordinates.

Note that this result is consistent with the fact that a 3D line has 4 degrees of freedom, since an element of $SO(2)$ has one degree of freedom and an element of $SO(3)$ has 3 degrees of freedom.

Using this representation of 3D lines, we show that there exists a locally non-singular minimal parameterization. Therefore, 3D lines can be locally updated with the minimum 4 parameters. The update scheme is inspired from those given above for 2D and 3D rotation matrices, and can be plugged into most of the existing non-linear optimization techniques. These results are summarized in table 3 in a practical manner.

Relating Plücker coordinates and the orthonormal representation. The orthonormal representation of a 3D line can be computed from its Plücker coordinates $\mathbf{L}^\mathsf{T} \sim (\mathbf{a}^\mathsf{T} \mid \mathbf{b}^\mathsf{T})$, as follows. Let $\bar{\mathsf{C}}_{(3\times 2)} \sim (\mathbf{a} \mid \mathbf{b})$ be factored as :

$$\bar{\mathsf{C}} \sim \underbrace{\begin{pmatrix} \frac{\mathbf{a}}{\|\mathbf{a}\|} & \frac{\mathbf{b}}{\|\mathbf{b}\|} & \frac{\mathbf{a}\times\mathbf{b}}{\|\mathbf{a}\times\mathbf{b}\|} \end{pmatrix}}_{SO(3)} \underbrace{\begin{pmatrix} \|\mathbf{a}\| & \\ & \|\mathbf{b}\| \end{pmatrix}}_{(\|\mathbf{a}\|\ \|\mathbf{b}\|)^\mathsf{T} \in \mathbb{P}^1}.$$

In practice, we use QR decomposition, $\bar{\mathsf{C}}_{(3\times 2)} = \mathsf{U}_{(3\times 3)}\Sigma_{(3\times 2)}$. As already mentioned the special form of matrix Σ is due to the Plücker constraint. While $\mathsf{U} \in SO(3)$, the two non-zero entries of Σ defined up to scale can be represented by an $SO(2)$ matrix W, as shown in §4.2.

Going back from the orthonormal representation to Plücker coordinates is trivial. The Plücker coordinates of the line are obtained from its orthonormal representation (U, W) as:

$$\mathbf{L}^\mathsf{T} \sim (w_{11}\mathbf{u}_1^\mathsf{T} \mid w_{21}\mathbf{u}_2^\mathsf{T}), \quad (15)$$

where $\mathbf{u}_i$ is the i-th column of U.

A 4-parameter update. Consider the orthonormal representation $(\mathsf{U}, \mathsf{W}) \in SO(3) \times SO(2)$ of a given 3D line. Since $\mathsf{U} \in SO(3)$, as reviewed in §4.2, it can not be minimally parameterized but can be locally updated using equation (11), as $\mathsf{U} \leftarrow \mathsf{UR}(\boldsymbol{\theta})$ where $\boldsymbol{\theta} \in \mathbb{R}^3$. Matrix $\mathsf{W} \in SO(2)$ can be updated as $\mathsf{W} \leftarrow \mathsf{WR}(\theta)$, where $\theta \in \mathbb{R}$. We define the update parameters by the 4-vector $\mathbf{p}^\mathsf{T} \sim (\boldsymbol{\theta}^\mathsf{T} \mid \theta)$.

Initialization. The initial guess is given by the Plücker coordinates $\mathbf{L}_0^\mathsf{T} \sim (\mathbf{a}_0^\mathsf{T} \mid \mathbf{b}_0^\mathsf{T})$.

- Compute the orthonormal representation $(\mathsf{U}, \mathsf{W}) \in SO(3) \times SO(2)$ of $\mathbf{L}_0$ by QR decomposition : $(\mathbf{a}_0 \mid \mathbf{b}_0) = \mathsf{U}\begin{pmatrix} \sigma_1 & \\ & \sigma_2 \end{pmatrix}$ and set $\mathsf{W} = \begin{pmatrix} \sigma_1 & -\sigma_2 \\ \sigma_2 & \sigma_1 \end{pmatrix}$.
- The 4 optimization parameters are $\mathbf{p}^\mathsf{T} = (\boldsymbol{\theta}^\mathsf{T} \mid \theta)$ where the 3-vector $\boldsymbol{\theta}$ and the scalar θ are used to update U and W respectively.

Update. (i.e. one optimization step)

- Current line is $\mathbf{L}^\mathsf{T} \sim (w_{11}\mathbf{u}_1^\mathsf{T} \mid w_{21}\mathbf{u}_2{}^\mathsf{T})$ and $\partial\mathbf{L}/\partial\mathbf{p}$ is given by equation (16).
- Compute $\mathbf{p}$ by minimizing some criterion.
- Update U and W: $\mathsf{U} \leftarrow \mathsf{UR}(\boldsymbol{\theta})$ and $\mathsf{W} \leftarrow \mathsf{WR}(\theta)$.

Table 3. Elements for 3D line optimization using the minimal 4 parameters through the orthonormal representation.

We denote J the (6×4) Jacobian matrix of the Plücker coordinates, with respect to the orthonormal representation. Matrix J must be evaluated at $\mathbf{p}_0 = \mathbf{0}_{(4\times 1)}$:

$$\mathsf{J}|_{\mathbf{p}_0} = \left(\left.\frac{\partial \mathbf{L}}{\partial \theta_1}\right|_{\mathbf{p}_0} \mid \left.\frac{\partial \mathbf{L}}{\partial \theta_2}\right|_{\mathbf{p}_0} \mid \left.\frac{\partial \mathbf{L}}{\partial \theta_3}\right|_{\mathbf{p}_0} \mid \left.\frac{\partial \mathbf{L}}{\partial \theta}\right|_{\mathbf{p}_0} \right).$$

By using the orthonormal representation to Plücker coordinates equation (15) and the Jacobian matrices for $SO(2)$ and $SO(3)$, as defined by equations (10,12,13,14), we obtain, after minor algebraic manipulations:

$$\mathsf{J}_{(6\times 4)} = \begin{pmatrix} \mathbf{0}_{(3\times 1)} & -\sigma_1\mathbf{u}_3 & \sigma_1\mathbf{u}_2 & -\sigma_2\mathbf{u}_1 \\ \sigma_2\mathbf{u}_3 & \mathbf{0}_{(3\times 1)} & -\sigma_2\mathbf{u}_1 & \sigma_1\mathbf{u}_2 \end{pmatrix}. \quad (16)$$

Geometric interpretation. Each of the 4 above-defined update parameters $\mathbf{p}$ has a geometric interpretation. Since matrix W encapsulates the distance d from the origin $\mathbf{O}$ to $\mathbf{L}$, parameter θ acts on d. Matrix U is related to a 3D coordinate frame attached to $\mathbf{L}$. Parameter θ_1 rotates $\mathbf{L}$ around a circle with radius d, centered on $\mathbf{O}$, and lying on the plane defined by $\mathbf{O}$ and $\mathbf{L}$. Parameter θ_2 rotates $\mathbf{L}$ around a circle with radius d, centered on $\mathbf{O}$, and lying in a plane containing $\mathbf{O}$, the closest point $\mathbf{Q}$ of $\mathbf{L}$ to $\mathbf{O}$, and perpendicular to $\mathbf{L}$. Parameter θ_3 rotates $\mathbf{L}$ around the axis defined by $\mathbf{O}$ and $\mathbf{Q}$. For the last three cases, the angles of rotation are the parameters themselves. This interpretation allows to easily incorporate a priori knowledge while estimating a line. For example, to leave the direction of the line invariant, one may use the 2 update parameters θ_2 and θ, while to leave the distance of the line to the origin invariant, one may use the 3 update parameters $\boldsymbol{\theta}$.

5. Experimental Results

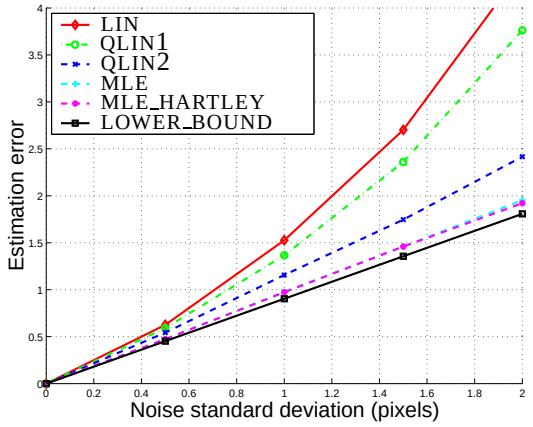
5.1. Simulated Data

Our simulated experimental setup consists of a set of cameras looking inwards at 3D lines randomly chosen in a sphere. Cameras are spread widely around the sphere. We fix the focal length of the cameras to 1000 (in number of pixels). Note that this information is not used in the rest of the experiments. The end-points of all lines are projected in all views, where their positions are corrupted by an additive Gaussian noise. We vary the parameters of this setup to assess and compare the quality of the different estimators on various scene configurations.

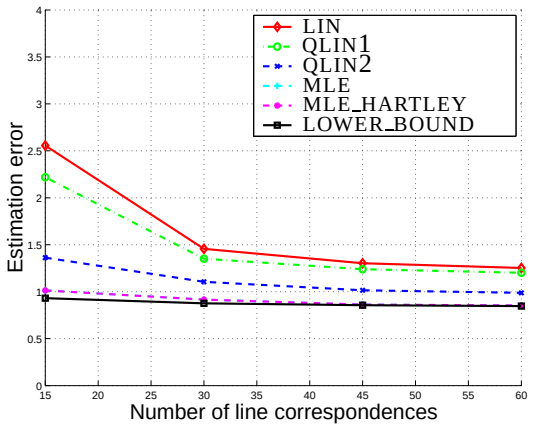
We compare the 4 methods given in this paper: LIN, QLIN1, QLIN2 and MLE (bundle adjustment based on our orthonormal representation of 3D lines), as well as the method given in [5, §15.4.1], denoted by 'MLE_HARTLEY'. This method consists in non-linearly computing the trifocal tensor as well as reconstructed lines by minimizing the reprojection error (4) and parameterizing the 3D lines by two of their three images. We also compare QLIN2 to a direct Levenberg-Marquardt-based minimization of the reprojection error: These two methods give undistinguishable results in all our experiments. Note that most existing methods, e.g. [6, 11, 12, 15] can be applied only when camera calibration is available.

We measure the quality of an estimate using the *estimation error*, as described in [5, §4], which also provides the theoretical lower bound. The estimation error is equivalent to the value of the negative log likelihood (4) (i.e. the reprojection error).

We vary the added noise level from 0 to 2 pixels, while considering 20 lines and 3 views. The result is shown on figure 1 (a). One observes that, beyond 1 pixel noise, methods



(a)



(b)

Figure 1. Estimation error for different methods when varying the variance of added noise on image end-points (a) and the number of lines considered (b).

LIN and QLIN1 behave very badly. This is mainly due to the bias introduced by the Plücker correction procedure. Methods QLIN2, MLE and MLE_HARTLEY degrade gracefully as the noise level increases. Method QLIN2 gives reasonable results. Methods MLE and MLE_HARTLEY give undistinguishable results, very close to the theoretical lower bound.

We vary the number of lines from 15 to 60, while considering a 1 pixel noise and 3 views. The result is shown on figure 1 (b). Similar conclusions as for the previous experiment can be drawn, except for the fact, that when more than 30 lines are considered, methods LIN and QLIN1 give reasonable results. Also, methods MLE and MLE_HARTLEY give results undistinguishable from the theoretical lower bound when more than 45 lines are considered.

Another experiment, not shown here due to lack of space, shows that when the number of views increases, the estimation error decreases for all compared methods. Note

that method MLE_HARTLEY can not be used with more than three views and is therefore not concerned with these conclusions. We observe that beyond 10 views, the result of method MLE is undistinguishable from the theoretical lower bound. The results given by methods LIN and QLIN1 are reasonable when more than 15 views are considered.

We observe that the quasi-linear methods always converge within 5 iterations.

5.2. Real Data

We tested our algorithms on several image sequences. For one of them, see figure 2, we show results. We provide

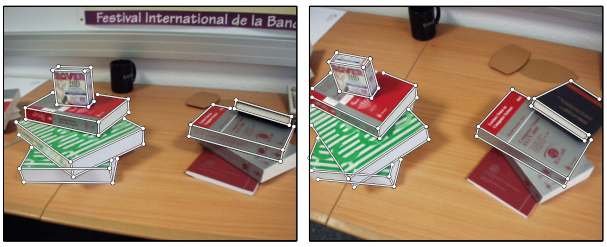


Figure 2. Sample images out of a 5-frame indoor sequence overlaid with manually-provided lines. Note that the optical distortion is not corrected.

45 line correspondences by hand. Note that some of them are visible in two views only. We use these line correspondences to compute the trifocal tensor corresponding to each subsequence formed by a triplet of consecutive images, using the linear method described in e.g. [5, §15.2]. We use method QLIN2 to reconstruct the lines associated with each triplet. We registered these subsequences by using the method given in [2]. At this point, we have a suboptimal guess of metric structure and motion. We further refine it using our structure from motion algorithms, to reconstruct each line by taking into account all of its images. The corresponding estimation errors are, respectively for LIN, QLIN1 and QLIN2, 2.3, 1.9 and 1.4 pixels.

We used the result of QLIN2 to initialize our maximum likelihood estimator for structure and motion based on the proposed orthonormal representation together with a metric parameterization of the camera motion, which ends up with a 0.9 pixel estimation error.

For each estimation, we reconstruct the end-points corresponding to the first view (shown on the left of figure 2). The maximum likelihood end-points are given by orthogonally projecting their images onto the image of the corresponding line.

These results are visible on figure 3. Note the significant improvement of methods QLIN2 and MLE over methods LIN

LIN & QLIN1

QLIN2

MLE

Figure 3. Zoom on some original (white) and reprojected lines (black).

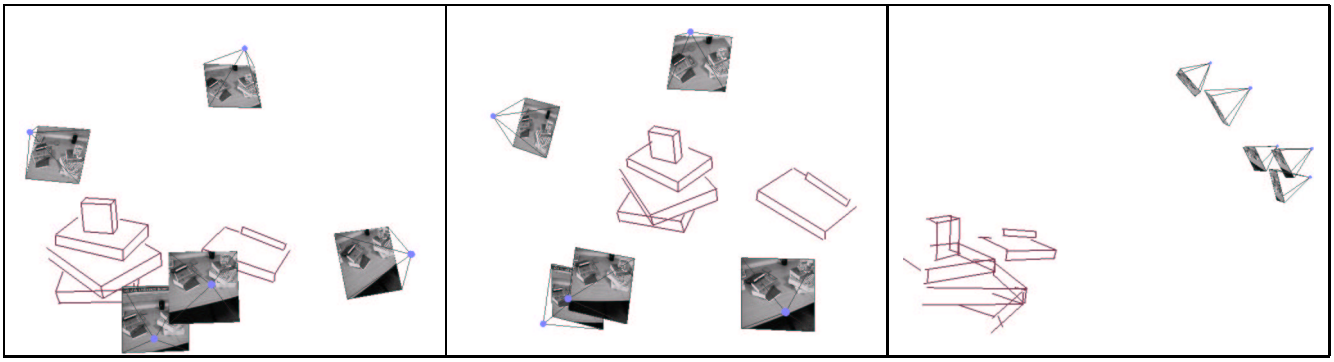


Figure 4. Snapshots of the cameras and lines reconstructed by method MLE. The images shown in figure 2 correspond to the top- and bottom-most cameras.

and QLIN1. The lines predicted by MLE and the original lines are undistinguishable. Figure 4 shows the cameras and lines reconstructed by MLE. There is visually no difference with the reconstruction provided by algorithm QLIN2, but that reconstructions provided by LIN and QLIN1 appear distorted.

6. Conclusion

We addressed the problem of structure and motion recovery from line correspondences across multiple views.

First, we proposed an optimal triangulation algorithm. Given camera motion, the Plücker coordinates of the lines are estimated by minimizing the reprojection error. The algorithm relies on an iteratively reweighted least squares scheme. We linearized the bilinear Plücker constraint to incorporate it up to first order in the estimation process. A Plücker correction procedure is proposed to find the nearest Plücker coordinates to a given 6-vector.

Second, we proposed the orthonormal representation of 3D lines, which allows non-linear optimization with the minimal 4 parameters within any unconstrained non-linear optimizer, contrarily to previously proposed overparameterizations. This representation is well-conditioned and allows analytic differentiation.

Experimental results on simulated and real data show that the standard linear method and its naive bias-corrected extension perform very badly in many cases and should only be used to initialize a non-linear estimator. Our bias-corrected algorithm including the Plücker constraint performs as well as direct Levenberg-Marquardt-based triangulation. It is therefore a good solution to initialize subsequent bundle adjustment. Based on our orthonormal representation, bundle adjustment gives results close to the theoretical lower bound and undistinguishable from the three-view maximum likelihood estimator of [5, §15.4.1], while being usable with any number of views.

Most algorithms proposed in this paper are summarized in a practical manner.

References

[1] 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.

[2] 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.

[3] 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.

[4] A. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed or open image sequences. In *European Conference on Computer Vision*, pages 311–326, june 1998.

[5] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, June 2000.

[6] Y. Liu and T. Huang. A linear algorithm for motion estimation using straight line correspondences. *Computer Vision, Graphics and Image Processing*, 44(1):35–57, October 1988.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] M. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4:171–183, 1990.

[12] C. Taylor and D. Kriegman. Structure and motion from line segments in multiple images. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 17(11):1021–1032, November 1995.

[13] 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.

[14] T. Viéville, Q. Luong, and O. Faugeras. Motion of points and lines in the uncalibrated case. *International Journal of Computer Vision*, 17(1), 1995.

[15] J. Weng, T. 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.