CVPR'03 — IN PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION, MADISON, WISCONSIN, USA, JUNE 2003.

Motion From 3D Line Correspondences: Linear and Non-Linear Solutions

Adrien Bartoli
INRIA Rhône-Alpes, France
Adrien.Bartoli@inria.fr

Richard I. Hartley
Australian National University
Richard.Hartley@anu.edu.au

Fredrik Kahl
Lund University, Sweden
fredrik@maths.lth.se

Abstract

We address the problem of aligning two reconstructions of lines and cameras in projective, affine, metric or Euclidean space. We propose several 3D and image-related linear algorithms. The result can be used to initialize the non-linear minimization of several proposed error functions, as well as the maximum likelihood estimator that we derive. We evaluate and compare our algorithms to existing ones using simulated and real data.

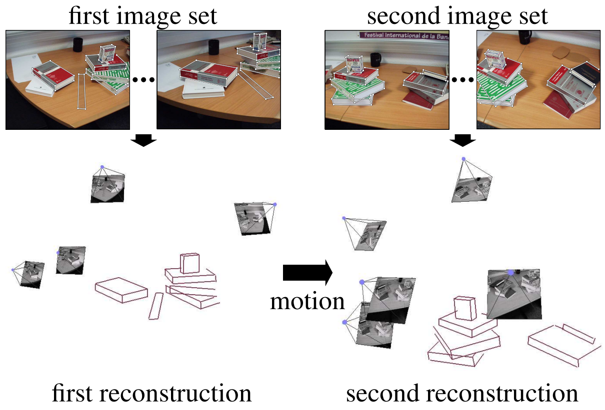

Figure 1. The problem addressed in this paper is the alignment of two reconstructions of cameras and lines.

1. Introduction

The problem of aligning two sets of 3D lines is considered. Lines are used in many areas of computer vision, e.g. in structure and motion estimation [12]. Suppose we are given two sets of images of the same scene. Each of these two sets gives rise to a 3D reconstruction of cameras and lines, expressed in its own basis of 3D space. Assuming that some of the reconstructed lines are shared between the two sets, it is possible to compute the transformation of 3D space that encapsulates the motion between the two reconstructions, as illustrated in Figure 1. Using this transformation, the two reconstructions can be expressed in the same 3D basis.

Our goals are to determine under which conditions one can compute the aligning transformation and to give algorithms for its estimation. These goals are sought for reconstructions expressed in projective, affine, metric and Euclidean spaces, i.e. when different levels of camera calibration and different camera models are considered.

While motion estimation from point correspondences has been well-studied [4, 11], its determination from line correspondences is still a challenging research area. There are three intrinsic difficulties to motion estimation from 3D line correspondences, which hold regardless the space considered. First, there is no global linear and minimal parameterization for using four global parameters. Second, there is no universally agreed error metric for comparing lines. Third, depending on the representation, it may be non-trivial to transfer a line between two different bases.

This paper has the following contributions. First, §4, we analyse the minimal cases, i.e. we determine, in each space considered, how many 3D line correspondences are necessary for motion estimation. Second, §5, we propose linear estimators which are capable of handling these minimal cases. It is also shown how point and line features can be directly incorporated. Third, §6, we propose non-linear estimators based on physically meaningful error functions. In particular, we derive the maximum likelihood estimator. Fourth, §7, we assess and compare our algorithms to existing ones, using simulated and real data. The following two sections give respectively some preliminaries and our notations and review the existing solutions.

2. Preliminaries and Notation

We make no formal distinction between coordinate vectors and physical entities. Equality up to a non-null scale factor is denoted by $\sim$, transposition and transposed inverse by $^\mathsf{T}$ 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. Everything is represented in homogeneous coordinates. Bars represent inhomogeneous leading parts of vectors or matrices, e.g. $\mathbf{M}^\mathsf{T} \sim \left(\bar{\mathbf{M}}^\mathsf{T} \mid m\right)$. We use $\|\mathbf{v}\|$ to designate the $\mathcal{L}_2$-norm of vector $\mathbf{v}$. The identity matrix is denoted I.

Scene settings. We consider two sets of reconstructed cameras and 3D lines. A different basis is attached to each of these two sets. Entities relative to the second set are followed by a prime. The first set contains n cameras, denoted by P_i, $i \in \{1,\ldots,n\}$. The second set contains n' cameras, denoted by $\mathsf{P}'_{i'}$, $i' \in \{1,\ldots,n'\}$. We assume that m line correspondences are given between the two reconstructions. Their Plücker coordinates (see below) are denoted by $\mathbf{L}_j \leftrightarrow \mathbf{L}'_j$, $j \in \{1,\ldots,m\}$.

Usual motion representation. A change of basis in projective, affine, metric and Euclidean spaces is conveniently represented by homogeneous (4×4) matrices, see e.g. [10, p.59]. The forms of these matrices are summarized in Table 1. In the general case, we use a (4×4) *generic motion matrix*, denoted by T.

projective	affine	metric	Euclidean
homography	affinity	similarity	rigidity
H, 15 dof	A, 12 dof	S, 7 dof	D, 6 dof
$\begin{pmatrix} \bar{\mathsf{H}} & \mathbf{h}_1 \\ \mathbf{h}_2^\mathsf{T} & h \end{pmatrix}$	$\begin{pmatrix} \bar{\mathsf{A}} & \mathbf{t} \\ \mathbf{0}_{(3)}^\mathsf{T} & 1 \end{pmatrix}$	$\begin{pmatrix} s\mathsf{R} & \mathbf{t} \\ \mathbf{0}_{(3)}^\mathsf{T} & 1 \end{pmatrix}$	$\begin{pmatrix} \mathsf{R} & \mathbf{t} \\ \mathbf{0}_{(3)}^\mathsf{T} & 1 \end{pmatrix}$

Table 1. Transformations in different spaces and their usual representations. Note that R denotes a 3D rotation.

Plücker line coordinates. Given two 3D points $\mathbf{M}^\mathsf{T} \sim (\bar{\mathbf{M}}^\mathsf{T} \mid m)$ and $\mathbf{N}^\mathsf{T} \sim (\bar{\mathbf{N}}^\mathsf{T} \mid n)$, one can form the Plücker coordinates of the line joining them as an homogeneous 6-vector $\mathbf{L}^\mathsf{T} \sim (\mathbf{a}^\mathsf{T} \mid \mathbf{b}^\mathsf{T})$ defined by $\mathbf{a} = \bar{\mathbf{M}} \times \bar{\mathbf{N}}$ and $\mathbf{b} = m\bar{\mathbf{N}} - n\bar{\mathbf{M}}$, see e.g. [10, p.53]. Note that other choices of constructing 6-vectors of Plücker coordinates are possible. Every choice goes with a bilinear constraint that 6-vectors have to satisfy in order to represent valid line coordinates. For our definition, the constraint is $\mathbf{a}^\mathsf{T}\mathbf{b} = 0$.

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 [1, 6] is given by:

$$\widetilde{\mathsf{P}} \sim (\det(\bar{\mathsf{P}})\bar{\mathsf{P}}^{-\mathsf{T}} \mid [\mathbf{p}]_\times \bar{\mathsf{P}}). \tag{1}$$

The 3D line motion matrices. Given a usual (4×4) motion matrix $\mathsf{T} \sim \begin{pmatrix} \bar{\mathsf{T}} & \mathbf{t}_1 \\ \mathbf{t}_2^\mathsf{T} & t \end{pmatrix}$, a (6×6) matrix called the 3D line motion matrix [1] is given by:

$$\widetilde{\mathsf{T}} \sim \begin{pmatrix} \det(\bar{\mathsf{T}})\bar{\mathsf{T}}^{-\mathsf{T}} & [\mathbf{t}_1]_\times \bar{\mathsf{T}} \\ -\bar{\mathsf{T}}[\mathbf{t}_2]_\times & t\bar{\mathsf{T}} - \mathbf{t}_1\mathbf{t}_2^\mathsf{T} \end{pmatrix}. \tag{2}$$

Corresponding lines $\mathbf{L} \leftrightarrow \mathbf{L}'$ represented by Plücker line coordinates are related by $\mathbf{L}' \sim \widetilde{\mathsf{T}}\mathbf{L}$.

3. Previous Work

In [1], methods for homography estimation based on the 3D line homography matrix $\widetilde{\mathsf{H}}$ are proposed. The first one, "3DLMM_LIN_3D", is based on a linear minimization of an algebraic distance between each $\mathbf{L}'$ and the corresponding $\widetilde{\mathsf{H}}\mathbf{L}$, over the 36 entries of $\widetilde{\mathsf{H}}$. The recovered $\widetilde{\mathsf{H}}$ does not therefore satisfy Equation (2) and subsequently it needs to be adjusted to represent a valid motion. One of the problems with this method is that, since 36 unknowns are involved, 7 line correspondences are necessary, which is more than the minimum 5 (see §4). The other linear methods given in [1] all require a minimum of 7 line correspondences.

Several non-linear methods are proposed in [1]. The most reliable one, "3DLMM_NLIN_2D", consists in minimizing the discrepancy between the reprojection of each $\widetilde{\mathsf{H}}\mathbf{L}$ in the second set of images and the corresponding image points defining the line.

Methods 3DLMM_LIN_3D and 3DLMM_NLIN_2D are compared to ours in §7.

4. Minimal Cases

We analyse minimal cases for motion estimation from 3D line correspondences, i.e. we seek the minimum number of line correspondences giving enough constraints to estimate the motion.

Since each line correspondence provides four constraints, it is necessary to have at least four lines to constrain the 15 degrees of freedom in a homography. However, that is not enough, an additional line is needed. This result is explained by the existence of a one-dimensional family of isotropies, mapping any group of four 3D lines onto itself [3, 5, 9]. This family of isotropies does not exist in the affine, metric and Euclidean spaces. Therefore, the minimal number of lines is straightforward to establish in these cases. They are summarized in Table 2.

projective	affine	metric	Euclidean
5	3	2	2

Table 2. Minimal cases for motion estimation from 3D line correspondences.

5. Linear Estimators

5.1. Deriving 3D Constraints

We show how to derive linear constraints on the generic motion matrix T. Consider one line correspondence $\mathbf{L} \leftrightarrow \mathbf{L}'$. The idea is to use a different representation for $\mathbf{L}$ and for $\mathbf{L}'$. In particular, we know that the usual motion representation is well-suited to the transfer of points or planes between bases. This suggests to represent $\mathbf{L}$ by a set of points $\mathbf{Q}_k \in \mathbf{L}$. The corresponding points $\mathbf{Q}'_k$ in the second basis can be written as a linear function (in homogeneous coordinates) of the motion matrix: $\mathbf{Q}'_k \sim \mathsf{T}\mathbf{Q}_k$. Since $\mathbf{L}$ and $\mathbf{L}'$ are corresponding, it follows $\mathbf{Q}'_k \in \mathbf{L}'$.

The goal is now to express these constraints by equations linear in the entries of T. For that purpose, let $\boldsymbol{\pi}'_l \ni \mathbf{L}'$ be a set of planes. We have $\mathbf{Q}'_k \in \boldsymbol{\pi}'_l$, which is algebraically expressed by:

$$(\mathsf{T}\mathbf{Q}_k)^\mathsf{T}\boldsymbol{\pi}'_l = 0, \tag{3}$$

which is linear in the entries of T. When r points $\mathbf{Q}_k \in \mathbf{L}$ and s planes $\boldsymbol{\pi}'_l \ni \mathbf{L}'$ are considered, we obtain a set of rs equations. Note that, as said in §4, at most 4 of them are independent.

Choosing the $\mathbf{Q}_k$ and the $\boldsymbol{\pi}'_l$. To use the constraint in (3), one needs to provide the points $\mathbf{Q}_k \in \mathbf{L}$ and the planes $\boldsymbol{\pi}'_l \ni \mathbf{L}'$. For example, the span representation of $\mathbf{L}$ provides two points, and similarly, the nullspace representation of $\mathbf{L}'$ provides two planes. It is clear that the more the $\mathbf{Q}_k$ are widely spread in the scene space, and the more the $\boldsymbol{\pi}'_l$ are well-distributed around the lines, the more stable the estimation will be. A possible choice is to use as $\mathbf{Q}_k$ the reconstructed image points, and as $\boldsymbol{\pi}'_l$ the viewing planes.

Using other features. Other feature correspondences, namely point and plane correspondences provide constraints on the motion. These constraints can be expressed as linear equations on the entries of the generic motion matrix T [4], and mixed with the above-derived constraints.

5.2. Deriving Image-Related Constraints

Following the previous derivation, we define a set of points $\mathbf{Q}_k \in \mathbf{L}$, and their correspondences $\mathbf{Q}'_k \sim \mathsf{T}\mathbf{Q}_k$ in the second basis. A similar goal as previously is sought: we want to express the constraints $\mathbf{Q}'_k \in \mathbf{L}'$ with equations linear in the entries of T. The difference is that we want these constraints to be image-based. For that purpose, we consider the second set of cameras $\mathsf{P}'_{i'}$. Since perspective projection preserves incidence relations, we have $\mathsf{P}'_{i'}\mathbf{Q}'_k \in \mathbf{l}'_{i'}$, where $\mathbf{l}'_{i'}$ is the image of $\mathbf{L}'$ into camera $\mathsf{P}'_{i'}$. This is algebraically expressed by:

$$\mathbf{l}'^{\mathsf{T}}_{i'}\mathsf{P}'_{i'}\mathsf{T}\mathbf{Q}_k = 0, \tag{4}$$

which is linear in the entries of T. When r points $\mathbf{Q}_k \in \mathbf{L}$ and n' images are considered, we obtain a set of rn' equations. The constraints (4) can be written as a linear system for the entries of the generic motion matrix T.

5.3. Homography Estimation

We show how an homography can be estimated based on the 3D or the image-related constraints previously derived.

5.3.1 Using 3D Constraints

One can replace the generic motion T in Equation (3) by a homography matrix H. Each line correspondence provides rs equations (r is the number of points $\mathbf{Q}_k \in \mathbf{L}$ and s is the number of planes $\boldsymbol{\pi}'_l \ni \mathbf{L}'$ considered). For m line correspondences, we derive the linear system $\Gamma_{(mrs\times 16)} \cdot \mathbf{h}_{(16)} = \mathbf{0}_{(mrs)}$, where:

$$\Gamma \sim \begin{pmatrix} & & \cdots & \\ \pi'_{l1}\mathbf{Q}_k^\mathsf{T} & \pi'_{l2}\mathbf{Q}_k^\mathsf{T} & \pi'_{l3}\mathbf{Q}_k^\mathsf{T} & \pi'_{l4}\mathbf{Q}_k^\mathsf{T} \\ & & \cdots & \end{pmatrix}, \tag{5}$$

and $\mathbf{h}$ is the row-wise vectorization of H, i.e. $\mathbf{h}^\mathsf{T} \sim (\mathbf{h}_1^\mathsf{T} \mid \mathbf{h}_2^\mathsf{T} \mid \mathbf{h}_3^\mathsf{T} \mid \mathbf{h}_4^\mathsf{T})$. In practice, due to noise in measurements, the rank of matrix Γ is 16. We minimize the error function $\|\Gamma\mathbf{h}\|^2$. The entries of matrix H are defined up to a free scale factor. We fix this free scale by adding the constraint $\|\mathbf{h}\|^2 = 1$. The solution for $\mathbf{h}$ is given by the singular vector associated to the smallest singular value of Γ [8].

Error analysis. Minimizing an error in projective space is not meaningful. The error measured is therefore purely algebraic. The previously described method consists of minimizing:

$$\|\Gamma\mathbf{h}\|^2 = \cdots + \|(\mathsf{H}\mathbf{Q}_k)^\mathsf{T}\boldsymbol{\pi}'_l\|^2 + \cdots.$$

This error metric depends on the free scales of points $\mathbf{Q}_k$ and planes $\boldsymbol{\pi}'_l$.

5.3.2 Using Image-Related Constraints

We replace the generic motion matrix T in Equation (4) by H. Each line correspondence provides rn' equations, where r is the number of points $\mathbf{Q}_k$ and n' the number of images. When m line correspondences are considered, we derive the linear system $\gamma_{(mrn'\times 16)} \cdot \mathbf{h}_{(16)} = \mathbf{0}_{(mrn')}$, where:

$$\gamma \sim \begin{pmatrix} & \cdots & & \\ \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'1}\mathbf{Q}_k^\mathsf{T} & \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'2}\mathbf{Q}_k^\mathsf{T} & \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'3}\mathbf{Q}_k^\mathsf{T} & \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'4}\mathbf{Q}_k^\mathsf{T} \\ & \cdots & & \end{pmatrix}. \tag{6}$$

As in the 3D case the rank of A is 16 and we minimize the error function $\|\gamma\mathbf{h}\|^2$ under the constraint $\|\mathbf{h}\|^2 = 1$. The solution is the singular vector of matrix γ associated to its smallest singular value.

Error analysis. By mimicking the reasoning of the 3D case, it follows that the error function minimized is

$$\|\gamma\mathbf{h}\|^2 = \cdots + \|\mathbf{l}'^{\mathsf{T}}_{i'}\mathsf{P}'_{i'}\mathsf{H}\mathbf{Q}_k\|^2 + \cdots.$$

Define the algebraic 2D point-to-line distance:

$$d_a^2(\mathbf{q}, \mathbf{l}) = (\mathbf{q}^\mathsf{T}\mathbf{l})^2. \tag{7}$$

The error function is equivalent to:

$$\sum_{j=1}^{j=m}\sum_{k=1}^{k=r}\sum_{i'=1}^{i'=n'} d_a^2(\mathsf{P}'_{i'}\mathsf{H}\mathbf{Q}_{kj}, \mathbf{l}'_{i'j}), \tag{8}$$

where $\mathbf{Q}_{kj} \in \mathbf{L}_j$ and $\mathbf{l}'_{i'j}$ is the image of $\mathbf{L}'_j$ on camera $\mathsf{P}'_{i'}$. This error function is not physically meaningful since the algebraic distance is biased compared to the Euclidean distance. In particular, it depends upon the homogeneous scale of its arguments.

5.4. Affinity Estimation

We show how to estimate an affinity based on the 3D or image-related constraints previously derived. Contrarily to the homography estimation the use of 3D constraints is physically meaningful.

5.4.1 Using 3D Constraints

Let A be an affinity matrix as defined in Table 1. We replace the generic motion matrix T by A into Equation (3). Provided that the $\mathbf{Q}^k$ are normalized such that $Q_4^k = 1$, we simplify the fourth term of the equation as $\pi'_{l4}\mathbf{Q}_k^\mathsf{T}\mathbf{a}_4 = \pi'_{l4}$. This leads to:

$$\pi'_{l1}\mathbf{Q}_k^\mathsf{T}\mathbf{a}_1 + \pi'_{l2}\mathbf{Q}_k^\mathsf{T}\mathbf{a}_2 + \pi'_{l3}\mathbf{Q}_k^\mathsf{T}\mathbf{a}_3 = -\pi'_{l4}.$$

Each line correspondence provides rs equations. Given m line correspondences, we form the linear system $\Phi_{(mrs\times 12)} \cdot \mathbf{a}_{(12)} = \mathbf{\Delta}_{(mrs)}$, where:

$$\Phi = \begin{pmatrix} & \cdots & \\ \boldsymbol{\pi}'_{l1}\mathbf{Q}_k^\mathsf{T} & \boldsymbol{\pi}'_{l2}\mathbf{Q}_k^\mathsf{T} & \boldsymbol{\pi}'_{l3}\mathbf{Q}_k^\mathsf{T} \\ & \cdots & \end{pmatrix} \text{ and } \mathbf{\Delta} = \begin{pmatrix} \cdots \\ -\pi'_{l4} \\ \cdots \end{pmatrix}$$

The 12-vector $\mathbf{a}$ is the row-wise vectorization of the 3 leading rows of A, i.e. $\mathbf{a}^\mathsf{T} \sim (\mathbf{a}_1^\mathsf{T} \mid \mathbf{a}_2^\mathsf{T} \mid \mathbf{a}_3^\mathsf{T})$. In practice, due to noise in measurements, the solution is not exact. We compute the least squares solution minimizing $\|\Phi\mathbf{a} - \mathbf{\Delta}\|^2$, through the pseudo-inverse of Φ.

Error analysis. As we did for the projective case, we derive the error measure associated to the above-described algorithm, as follows:

$$\|\Phi\mathbf{a} - \mathbf{\Delta}\|^2 = \cdots + \|(\mathsf{A}\mathbf{Q}_k)^\mathsf{T}\boldsymbol{\pi}'_l\|^2 + \cdots.$$

This expression is similar to that obtained for the estimation of an homography. However, provided that some normalization constraints are satisfied by the data, this error function is physically meaningful. Indeed, each term can be interpreted as the squared orthogonal (Euclidean) point-to-plane distance. The normalization constraints are $Q_4^k = 1$, as already required above, and $\pi'^2_{l1} + \pi'^2_{l2} + \pi'^2_{l3} = 1$.

5.4.2 Using Image-Related Constraints

We replace the generic motion matrix T in Equation (4) by A. Similarly to the 3D case, we assume $Q_4^k = 1$, which allows to simplify the fourth term of the equation as $\mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'4}\mathbf{Q}_k^\mathsf{T}\mathbf{a}_4 = \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'4}$. This leads to:

$$\mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'1}\mathbf{Q}_k^\mathsf{T}\mathbf{a}_1 + \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'2}\mathbf{Q}_k^\mathsf{T}\mathbf{a}_2 + \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'3}\mathbf{Q}_k^\mathsf{T}\mathbf{a}_3 = -\mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'4}.$$

Each line correspondence provides rn' equations. Given m line correspondences, we form the linear system $\phi_{(mrn'\times 12)} \cdot \mathbf{a}_{(12)} = \delta_{(mrn')}$, where:

$$\phi = \begin{pmatrix} & \cdots & \\ \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'1}\mathbf{Q}_k^\mathsf{T} & \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'2}\mathbf{Q}_k^\mathsf{T} & \mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'3}\mathbf{Q}_k^\mathsf{T} \\ & \cdots & \end{pmatrix} \tag{9}$$

$$\text{and } \delta = \begin{pmatrix} \cdots \\ -\mathbf{l}'^{\mathsf{T}}_{i'}\mathbf{p}'_{i'4} \\ \cdots \end{pmatrix}.$$

As in the 3D case, due to noise, the solution is not exact. We minimize the error function $\|\phi\mathbf{a} - \delta\|^2$. The solution is obtained using the pseudo inverse of matrix ϕ.

Error analysis. We follow the same reasoning as for the estimation of an homography based on the image-related constraints, see §5.3.2. We obtain the same result, i.e. the error function is given by Equation (8). This is a sum of squared algebraic distances between reprojected points and measured images lines. Contrarily to the 3D case, the error function is not physically meaningful, for the reasons explained in §5.3.2.

5.5. Similarity and Rigidity Estimation

This case is different from the previously-studied affinity and homography cases, in that the motion that will be recovered includes a rotation matrix, see matrices S and D in Table 1. Hence, non-linear constraints must hold, namely the orthonormality of the rotation matrix. Such constraints are not trivial to handle.

5.5.1 Solution For $m \geq 3$ Line Correspondences

One possibility is to compute an affinity using one of the previously given methods. The leading (3×3) block $\bar{\mathsf{A}}$ can then be corrected to be a rotation matrix (up to scale in the similarity case), as follows. Let $\bar{\mathsf{A}} = \mathsf{U}\mathrm{diag}(\sigma_1, \sigma_2, \sigma_3)\mathsf{V}^\mathsf{T}$ be a singular value decomposition of $\bar{\mathsf{A}}$. The scale factor is given by $s = (\sigma_1 + \sigma_2 + \sigma_3)/3$. The rotation is given by $\mathsf{R} = \mathsf{U}\mathsf{V}^\mathsf{T}$. This correction guarantees that $s\mathsf{R}$ is the closest scaled rotation matrix to $\bar{\mathsf{A}}$, in the sense of the Frobenius norm.

Note that three line correspondences are necessary to compute the affinity, one more than the minimum number in the similarity and rigidity cases (see Table 2).

5.5.2 Solution For $m \geq 2$ Line Correspondences

The idea is to use the direction of lines to compute the rotational part of the transformation. We consider the 3D line similarity matrix $\widetilde{\mathsf{S}}$ relating the Plücker coordinates of corresponding lines by:

$$\lambda_j \mathbf{L}'_j = \widetilde{\mathsf{S}}\mathbf{L}_j, \tag{10}$$

where λ_j is a non-zero scale factor and $\widetilde{\mathsf{S}} \sim \begin{pmatrix} s\mathsf{R} & [\mathbf{t}]_\times \mathsf{R} \\ 0_{(3\times 3)} & \mathsf{R} \end{pmatrix}$. By expanding Equation (10), we obtain:

$$\begin{aligned} \lambda_j \mathbf{a}'_j &= s\mathsf{R}\mathbf{a}_j + [\mathbf{t}]_\times \mathsf{R}\mathbf{b}_j & (11) \\ \lambda_j \mathbf{b}'_j &= \mathsf{R}\mathbf{b}_j. & (12) \end{aligned}$$

Computing R and the λ_j. We compute the rotation by using only the directional part of the lines, contained in the bottom 3-vector of the Plücker coordinates. The corresponding Equation (12) shows that indeed, the transformation from $\mathbf{b}_j$ to $\mathbf{b}'_j$ is not affected by the scale and the translation. By taking the norm of each side of Equation (12), we obtain $\|\lambda_j \mathbf{b}'_j\| = \|\mathsf{R}\mathbf{b}_j\|$, which allows to compute the magnitude of the λ_j as $|\lambda_j| = \|\mathbf{b}_j\|/\|\mathbf{b}'_j\|$. Note that $\|\mathbf{b}_j\|$ (or equivalently $\|\mathbf{b}'_j\|$) is zero if and only if the corresponding line lies on the plane at infinity. Since only the magnitude of λ_j is determined, $\mathbf{L}_j$ and $\mathbf{L}'_j$ need to be normalized so that all the direction vectors $\mathbf{b}_j$ and $\mathbf{b}'_j$ lie on the same side of the unit sphere. To compute the rotation, we wish to minimize:

$$\sum_{j=1}^{j=m} d^2(\mathbf{b}'_j, \mathsf{R}\mathbf{b}_j), \qquad (13)$$

where d is the Euclidean distance. In [11], the author demonstrates that the solution to this problem is given by the closest orthonormal matrix to $\mathsf{M}_{(3\times 3)} = \sum_{j=1}^{j=m} \mathbf{b}'_j \mathbf{b}_j^\mathsf{T}$. Let $\mathsf{M} = \mathsf{U}\Sigma\mathsf{V}^\mathsf{T}$ be a singular value decomposition of matrix M, then $\mathsf{R} = \mathsf{U}\mathsf{V}^\mathsf{T}$ gives the required solution.

Computing $\mathbf{t}$ and s. Consider Equation (11). It can be transformed into the linear system $\mathsf{E}_{(3m\times 4)} \cdot \mathbf{f}_{(4)} = \mathbf{g}_{(3m)}$, where:

$$\mathsf{E} = \begin{pmatrix} & \cdots & \\ \mathsf{R}\mathbf{a}_j & & [\mathsf{R}\mathbf{b}_j]_\times^\mathsf{T} \\ & \cdots & \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} s \\ \mathbf{t} \end{pmatrix} \text{ and } \mathbf{g} = \begin{pmatrix} \cdots \\ \lambda_j \mathbf{a}'_j \\ \cdots \end{pmatrix}.$$

We compute the solution that minimizes $\|\mathsf{E}\mathbf{f} - \mathbf{g}\|^2$ through the pseudo-inverse of matrix E.

Error analysis. The error functions considered in the two parts of this algorithm are not meaningful since 3D lines are used directly (as said in the introduction, there does not exist a physically meaningful metric between 3D lines in the general case). For example, in the first part of the algorithm, we minimize the error function (13), which is not physically meaningful in the sense that line directions, i.e. points lying at infinity, are compared. This method is useful in the minimal case, i.e. when two line correspondences only are avaible. It can used e.g. to bootstrap a random sampling-based robust estimator such as RANSAC [7]. The method also has the advantage that the orthogonality constraints are enforced directly and not afterwards. However, when three line correspondences or more are available, the experimental results indicate that the method of §5.5.1 is superior.

6. Non-Linear Estimators

Most of the above-proposed linear estimators are not based on physically meaningful error functions. For that reason, we propose several physically meaningful error functions.

6.1. Distances in One Set of Images

The error function (8) is based on an algebraic 2D point-to-line distance. We replace the algebraic distance by the orthonormal (Euclidean) point-to-line distance defined by $d_\perp^2(\mathbf{q}, \mathbf{l}) = \frac{(\mathbf{q}^\mathsf{T}\mathbf{l})^2}{q_3^2(l_1^2+l_2^2)}$. This yields the following error function:

$$\sum_{j=1}^{j=m} \sum_{k=1}^{k=r} \sum_{i'=1}^{i'=n'} d_\perp^2(\mathsf{P}'_{i'}\mathsf{T}\mathbf{Q}_{kj}, \mathbf{l}'_{i'j}). \qquad (14)$$

We propose two effective methods to minimize it.

6.1.1 Levenberg-Marquardt Optimization

The error function (14) is a non-linear least squares function. We use a Newton-type algorithm, namely Levenberg-Marquardt, see e.g. [8] to minimize it over the motion parameters.

6.1.2 Quasi-Linear Optimization

Consider the error functions (8) and (14). The former can be minimized linearly, while the latter needs non-linear optimization. They are respectively based on the algebraic distance d_a and the Euclidean distance $d_\perp$, that can be related by $w^2 d_\perp^2(\mathbf{q}, \mathbf{l}) = d_a^2(\mathbf{q}, \mathbf{l})$, where the weight factor w is given by $w^2 = q_3^2(l_1^2 + l_2^2)$. Since the lines considered in Equations (8,14) are a priori known, they can be normalized such that $l_1^2 + l_2^2 = 1$. We obtain therefore:

$$w_{i'kj}^2 = \mathbf{p}'^{\mathsf{T}}_{i'3} \mathsf{T}\mathbf{Q}_{kj}, \qquad (15)$$

where $\mathbf{p}'^{\mathsf{T}}_{i'3}$ is the third row of matrix $\mathsf{P}'_{i'}$. Such weight factors encapsulate the bias of the linear versus the non-linear method. The fact that they depend upon the unknown motion matrix T suggests the following iterative scheme:

1. *Initialization:* set $w_{i'kj} = 1$, $\forall i' \forall k \forall j$, and form the linear system γ from Equation (6) for the projective case or λ, δ from Equation (9) for the other cases.

2. *Estimation:* estimate the motion matrix by solving the linear system, using the appropriate method (see §§5.3.2 and 5.4.2 for the projective and the other cases respectively). If necessary (i.e. in the metric and Euclidean cases), correct the recovered matrix to enforce the orthonormality constraints (see §5.5.1).

3. *Bias-correction:* use the computed motion to estimate the weights $w_{i'jk}$ from Equation (15) and reweight the linear system considered.

4. *Iteration:* iterate steps 2 and 3 until convergence (see main text).

Convergence is determined by thresholding the difference between two consecutive errors. It is reached in typically 3 to 5 iterations.

6.2. Distances in Both Sets of Images

The error function in (14) is not symmetric in that the two sets of images do not play the same role. We propose a similar error function, but expressed in a symmetric manner, as follows:

$$\sum_{j=1}^{j=m}\sum_{k=1}^{k=r}\left(\sum_{i'=1}^{i'=n'} d_\perp^2(\mathsf{P}'_{i'}\mathsf{T}\mathbf{Q}_{kj}, \mathbf{l}'_{i'j}) + \sum_{i=1}^{i=n} d_\perp^2(\mathsf{P}_i\mathsf{T}^{-1}\mathbf{Q}'_{kj}, \mathbf{l}_{ij})\right), \quad (16)$$

where the P_i, $i \in \{1,\ldots,n\}$ are the projection matrices related to the first set of cameras, points $\mathbf{Q}'_{kj} \in \mathbf{L}'_j$ and $\mathbf{l}_{ij}$ is the image of $\mathbf{L}_j$ in camera P_i.

We propose to minimize this non-linear least-squares error function as in the previous case, using the Levenberg-Marquardt algorithm.

6.3. Maximum Likelihood Estimation

We derive the error function corresponding to the maximum likelihood estimator. Contrarily to previously given methods, the optimization must be conducted over both the motion parameters and corrected line positions, that perfectly match through the recovered motion.

To derive the maximum likelihood estimator, we make the following assumptions. First, we assume that each measured image line, $\mathbf{l}_{ij}$ in the first set of images or $\mathbf{l}'_{i'j}$ in the second set, has been obtained as the best fit to a set of two or more points. This is the case when the user specifies the end-points of the line, or when the line is automatically detected, as in e.g. [2]. In [10, p.393], it is shown that the best fit to a set of multiple points can be brought back to the two point case. Without loss of generality, we denote $\mathbf{x}_{ij}$ and $\mathbf{y}_{ij}$ the two points defining $\mathbf{l}_{ij}$. A similar notation holds for the points corresponding to $\mathbf{l}'_{i'j}$.

Second, we assume that the position of these points is corrupted from their underlying true position by an additive Gaussian noise. We assume that this noise is i.i.d. (identically and independently distributed).

Under these assumptions, the maximum likelihood error function is obtained as follows. Let $\widehat{\mathbf{L}}_j$ be the corrected lines, expressed in the first basis. These lines are reprojected into the first and the second set of images. The sum of squared orthogonal distances between these reprojected lines and the corresponding image points is minimized, both over corrected line positions and the motion parameters. More formally, the optimization problem to be solved is:

$$\min_{\mathsf{T},\widehat{\mathbf{L}}_1,\ldots,\widehat{\mathbf{L}}_m} \sum_{j=1}^{j=m}\left(\sum_{i=1}^{i=n}\left(d_\perp^2(\mathbf{x}_{ij}, \widetilde{\mathsf{P}}_i\widehat{\mathbf{L}}_j) + d_\perp^2(\mathbf{y}_{ij}, \widetilde{\mathsf{P}}_i\widehat{\mathbf{L}}_j)\right) + \sum_{i'=1}^{i'=n'}\left(d_\perp^2(\mathbf{x}'_{i'j}, \widetilde{\mathsf{P}}'_{i'}\widetilde{\mathsf{T}}\widehat{\mathbf{L}}_j) + d_\perp^2(\mathbf{y}_{i'j}, \widetilde{\mathsf{P}}'_{i'}\widetilde{\mathsf{T}}\widehat{\mathbf{L}}_j)\right)\right). \quad (17)$$

In this equation, notations $\widetilde{\mathsf{P}}_i$ and $\widetilde{\mathsf{P}}'_{i'}$ designate the perspective projection matrices for Plücker line coordinates, see Equation (1), associated to P_i and $\mathsf{P}'_{i'}$ respectively. Notation $\widetilde{\mathsf{T}}$ is the 3D line motion matrix associated to T, see Equation (2).

Non-linear optimization over 3D lines is not trivial. We optimize two image lines representing each 3D line, as suggested in [10, p.386]. The Levenberg-Marquardt algorithm is used to conduct the optimization.

7. Experimental Results

7.1. Simulated Data

Our experimental setup consists of two sets of $n = n' = 5$ cameras observing 3D lines randomly chosen inside a sphere. To simulate realistic images, we fix the focal length of the cameras to 1000 (expressed in number of pixels). Note that this information is not used in the rest of the experiments. The end-points of all lines are projeted in all views, where their positions are corrupted by an additive Gaussian noise. The camera matrices are set in the coordinate frame defined by the first camera of each set. The lines of each set of cameras are independently reconstructed by minimizing their reprojection error. As said previously, camera calibration is not used. The reconstructions are therefore expressed in projective space. We vary separately the parameters of this setup to assess and compare the quality of the proposed estimators to existing ones on various scene configurations.

We briefly describe the 10 compared methods.

Method PTS_LIN_3D and PTS_NLIN_2D_SYM, [4] are based on point correspondences. The former consists in a linear estimation of the motion parameters, based on a 3D algebraic error between points transferred from the first basis to the second one. The latter consists in a non-linear minimization of the discrepancy between reprojected points, transferred between bases and reprojected onto the original images.

Methods 3DLMM_LIN_3D and 3DLMM_NLIN_2D based on the 3D line motion matrix are described in §3.

For methods LIN_3D, LIN_2D, NLIN_2D, QLIN_2D, NLIN_2D_SYM and MLE, the reader is referred to, respectively, §§5.3.1, 5.3.2, 6.1.1, 6.1.2, 6.2 and 6.3.

Non-linear estimators are initialized with the result given by LIN_2D. Note that image point and line coordinates are standardized, such that the image corners lie at $(\pm 1, \pm 1, 1)^{\mathsf{T}}$.

We assess the quality of an estimated motion by measuring its *estimation error*, as described in [10, p.117]. The idea is to correct the reconstructed lines so that they perfectly satisfy the recovered motion. The reprojection er-

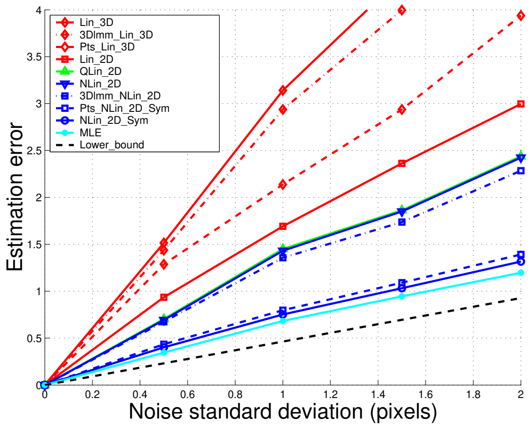

Figure 2. Estimation error for different methods when varying the variance of added noise on image points.

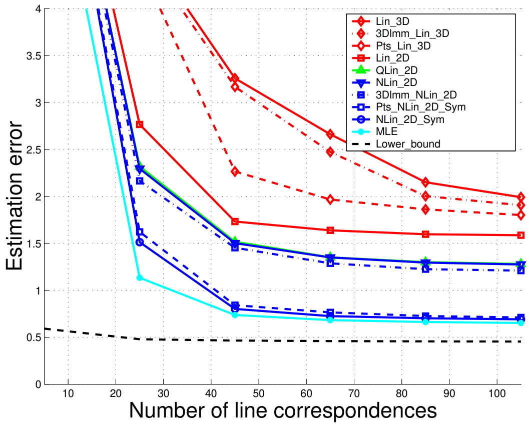

Figure 3. Estimation error for different methods when varying the number of simulated line correspondences.

ror, see Equation (17), as minimized by the maximum likelihood estimator, gives the estimation error. This quality measure reflects how much the error in the data has been reduced by estimating the underlying motion parameters. The framework of [10, p.117] provides the associated theoretical lower bound.

The first experiment is based on varying the variance of added noise on image point position, from 0 to 2 pixels. Note that a 2 pixel error is more than should occur with a careful positionning of lines on real images, in particular when lines are fitted to several points. The number of line correspondences used in this experiment is $m = 50$. We observe in Figure 2 that the estimation error degrades gracefully as the added noise level on the image points increases. This is true for all tested methods.

The second experiment is based on changing the number m of simulated lines, from 5 to 105. Note that, as stated in §4, 5 is the minimum number of line correspondences for projective motion estimation. On the other hand, 20-100 is a number of lines that one might expect to observe in real images, depending on the scene considered. We observe in Figure 3 that most methods do not reduce the estimation error beyond a number of 50 line correspondences. This is not true for methods based on a 3D error function, which seem to stabilize their results only beyond 90 line correspondences.

The following remarks hold for both experiments.

- Lin_3D, 3Dlmm_Lin_3D and Pts_Lin_3D are highly sensitive to noise and should not be used.
- Lin_2D provides results reliable enough to initialize subsequent non-linear estimators. It is simple to implement and runs with minimal and redundent data. It is therefore well-suited to bootstrap random sampling-based estimators, such as RANSAC [7].
- QLin_2D, NLin_2D and 3Dlmm_NLin_2D, give very similar results. Note that method QLin_2D is implemented as a reweighting loop over Lin_2D.
- Pts_NLin_2D_Sym, NLin_2D_Sym and MLE give slightly different results. When computational cost is relevant, NLin_2D_Sym should be preferred.

7.2. Real Data

We consider the two sets of respectively 4 and 5 images and 40 and 45 line correspondences from which samples are shown in Figure 1. We reconstruct the cameras and the lines in metric space and apply our algorithms to align the two reconstructions, using 21 line correspondences. Note that these correspondences are closely located in space, thereby making the alignment task unstable.

Figures 1 and 4 (a) show the initial reconstructions, i.e. before alignment. Figure 4 (b) shows the result obtained using the algorithm of §5.5.2, based on an algebraic error function. Figures 4 (c) and 5 show the result obtained using the algorithm of §5.5.1, based on a physically meaningful error function.

The results given by the latter algorithm are clearly much better. The reprojection errors (in the second set of images) are respectively 15.3 and 3.7 pixels.

8. Conclusions

We tackled the problem of motion computation from 3D line correspondences. We analysed the minimal cases and gave linear methods for various spaces. Error functions expressed in 3D or image-based are considered. Several

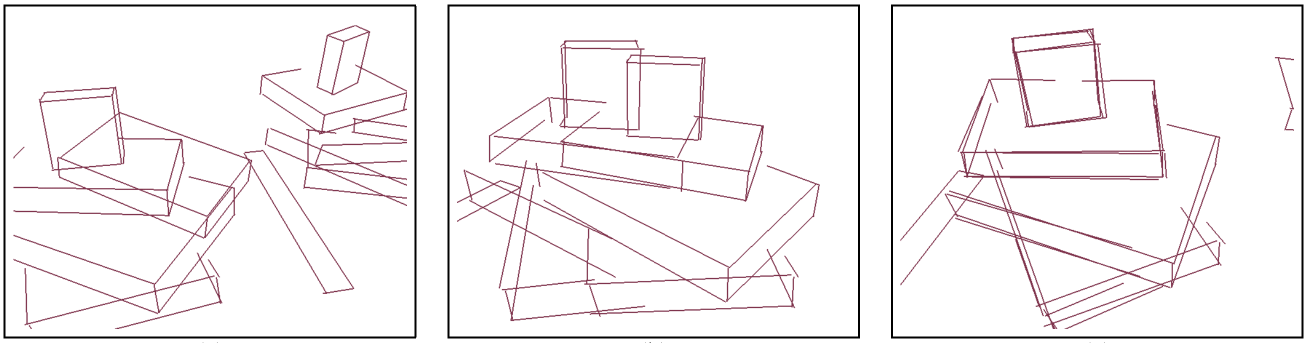
(a) (b) (c)

Figure 4. The reconstructions without alignment (a) and aligned with the algorithms of §§5.5.2 (b) and 5.5.1 (c).

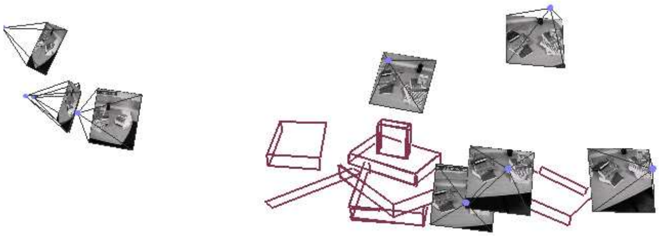

Figure 5. The two reconstructions aligned by computing an affinity and correcting it so that it represents a rigid displacement, see §5.5.1.

non-linear estimators are proposed, including the maximum likelihood estimator for motion and reconstructed lines.

We evaluated our algorithms on simulated and real data, including comparison with existing ones and to point-based methods.

References

[1] 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.

[2] J. Canny. A computational approach to edge detection. IEEE *Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[3] H. Crapo and R. Penne. Cheirality and the isotropy classification of skew lines in projective 3-space. *Advances in Mathematics*, 103(1):1–106, 1994.

[4] G. Csurka, D. Demirdjian, and R. Horaud. Finding the collineation between two projective reconstructions. *Computer Vision and Image Understanding*, 75(3):260–268, September 1999.

[5] G. Csurka and O. Faugeras. Computing three-dimensional projective invariants from a pair of images using the Grassmann-Cayley algebra. In *Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision, Xian, China*, pages 150–157, 1995.

[6] 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.

[7] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Graphics and Image Processing*, 24(6):381 – 395, June 1981.

[8] P. Gill, W. Murray, and M. Wright. *Practical Optimization*. Academic Press, 1981.

[9] R. Hartley. Invariants of lines in space. In *Proceedings of* DARPA *Image Understanding Workshop*, pages 737–744, 1993.

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

[11] B. Horn, H. Hilden, and S. Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7):1127–1135, July 1988.

[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.