Estimating Motion and Structure from Correspondences of Line Segments Between Two Perspective Images

Zhengyou Zhang

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Estimating Motion and Structure from Correspondences of Line Segments Between Two Perspective Images

Zhengyou Zhang

N° 2340

September 1994

PROGRAMME 4

Robotique,

image

et vision

ISSN 0249-6399

Estimating Motion and Structure from Correspondences of Line Segments Between Two Perspective Images

Zhengyou Zhang

Programme 4 — Robotique, image et vision
Projet Robotvis

Rapport de recherche n ° 2340 — September 1994 — 34 pages

Abstract: We present in this paper an algorithm for determining 3D motion and structure from correspondences of *line segments* between two perspective images. To our knowledge, this paper is the first investigation on use of line segments in motion and structure from motion. Classical methods use their geometric abstraction, namely straight lines, but then three images are necessary. We show in this paper that two views are in general sufficient when using line segments. The assumption we use is that two matched line segments contain the projection of a *common part* of the corresponding line segment in space. Indeed, this is what we use to match line segments between different views. Both synthetic and real data have been used to test the proposed algorithm, and excellent results have been obtained with real data containing about one hundred line segments. The results are comparable with those obtained with calibrated stereo.

Key-words: Motion, Structure from Motion, Line Segments, Epipolar Geometry, Overlap

(Résumé : tsvp)

Unité de recherche INRIA Sophia-Antipolis
2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex (France)
Téléphone : (33) 93 65 77 77 – Télécopie : (33) 93 65 77 65

Estimation du mouvement et de la structure à partir de segments de droite entre deux images perspectives

Résumé : Dans cet article, nous présentons un algorithme permettant d'estimer le mouvement et la structure à partir d'appariements de *segments de droite* entre deux images perspectives. À notre connaissance, cet article est la première tentative d'utilisation de segments de droite à cette fin. Contrairement aux méthodes usuelles qui utilisent les droites support de ces segments (pour lesquelles trois vues au minimum sont nécessaires), nous montrons dans cet article que deux vues sont en général suffisantes avec les segments de droite. L'hypothèse cruciale qui entre en compte est que deux segments de droite appariés contiennent la projection d'une partie commune d'un même segment de l'espace. Cela est en fait déjà utilisé pour la mise en correspondance de segments entre plusieurs images. L'algorithme proposé a été testé avec des données synthétiques et réelles, et d'excellents résultats ont été obtenus avec des données réelles contenant une centaine de segments. Les résultats sont comparables avec ceux obtenus avec la stéréo calibrée.

Mots-clé : Mouvement, Structure à partir du mouvement, Segments de droite, Géométrie épipolaire, Recouvrement

Contents

List of Figures

1 Introduction

The problem of estimating motion and structure from two or three images has been studied for a while in the computer vision community. We can trace it back to the late seventies: Ullman [19] assumed a orthographic camera projection model and showed that three views are necessary to recover the motion and structure from point correspondences; Roach and Aggarwal [15] used a full perspective projection model and thus two views are sufficient from point correspondences. Since then, many approaches have been proposed to solve the problem using either linear or nonlinear methods. The reader is referred to [11] for a complete review.

Essentially, two types of geometric primitives have been used in solving motion and structure problem, namely points and straight lines. When points are used, two perspective views are sufficient to recover the motion and structure of the scene. When straight lines are used, three perspective views are necessary. Closed-form solutions are available either for point correspondences [13, 18] or for line correspondences [17, 12]. Algorithms using both points and lines are also available [16]. However, another important type of geometric primitives, namely that of *line segments*, has been since long ignored in motion and structure from motion[1]. The importance of line segments in computer vision has never been underestimated. As a matter of a fact, straight lines are merely the geometric abstraction of line segments by ignoring their endpoints. The overlook of line segments in the domain of motion and structure from motion is probably due to the lack of mathematical elegance in representing line segments.

To our knowledge, this paper is the first investigation in computer vision on motion and structure from correspondences of line segments. Unlike the case of straight lines, we show that two views are generally enough to recover the motion and structure of the scene. The only assumption we use is that two matched line segments contain the projection of a *common part* of the corresponding line segment in space (and we say that the two 2D line segments overlap). Indeed, this assumption is minimal, and is what we use to match line segments between different views.

We do not address the problem of matching line segments here. This can be done by tracking [4, 8] or other techniques [3]. This paper is organized as follows. Section 2 describes the problem we want to solve and shows why we can determine 3D motion and structure from corresponding line segments between two images. Section 3 presents the algorithm for solving the motion problem. The epipolar constraint is first described and the concept of overlap between two matched line segments is then introduced based on the epipolar constraint. The motion problem is finally solved by maximizing the overlap between two sets of lines segments. Section 4 addresses the issue of 3D reconstruction of line segments

[1] 3D line segments, reconstructed by a stereo system, have been used in motion analysis by Zhang and Faugeras [21], but the problem there is different from the one addressed here.

provided the motion is estimated. Section 5 provides the experimental results with both synthetic and real data.

2 Statement of the Problem

In this section, we describe the geometry of line segments in motion, introduce the minimal amount of notation required, and define the problem we want to solve.

2.1 Notation

We use bold low case letters $\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$, $\cdots$ for column vectors and for points in image plane, capital letters A, B, C, $\cdots$ for points in 3D space, and bold capital letters $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\cdots$ for matrices. The superscript ^{T} denotes the transpose of a vector or a matrix, and thus $\mathbf{a}^T$ is the row vector corresponding to $\mathbf{a}$, and $\mathbf{A}^T$ is the transpose of $\mathbf{A}$. The cross product of two vectors $\mathbf{a}$ and $\mathbf{b}$ is denoted by $\mathbf{a} \times \mathbf{b}$. The dot product of $\mathbf{a}$ and $\mathbf{b}$ is denoted by $\mathbf{a} \cdot \mathbf{b}$ or $\mathbf{a}^T\mathbf{b}$.

The coordinates of a point $\mathbf{a}$ in image plane are $[u, v]^T$; its homogeneous coordinates are $[u, v, 1]^T$, and are denoted by $\widetilde{\mathbf{a}}$. Similarly, for a point $M = [x, y, z]^T$ in 3D space, its homogeneous coordinates are $[x, y, z, 1]^T$, and are denoted by $\widetilde{M}$.

Additional notation will be introduced in the following subsections.

2.2 Geometry of the motion problem of line segments

We model our camera as a standard pinhole. The relation between each point M in space and its corresponding point $\mathbf{m}$ in image plane is linear projective, and is described by a perspective transformation, i.e.,

$$s\widetilde{\mathbf{m}} = \mathbb{P}\widetilde{M} \,, \tag{1}$$

where s is an arbitrary scale and $\mathbb{P}$ is a 3×4 matrix known as the perspective projection matrix. To each camera is associated a coordinate frame $Cy_1y_2y_3$ (see Fig. 1), in which the positions of the image points are measured. The optical center of the camera is at C. The optical axis is aligned with the y_3 axis. The image plane is parallel to the y_1y_2 plane and is at $y_3 = 1$ (i.e. the focal length is equal to 1). Thanks to camera calibration, $\mathbb{P}$ has the following simple form:

$$\mathbb{P} = [\mathbf{R}_{wc},\ \mathbf{t}_{wc}] \ ,$$

where $\mathbf{R}_{wc}$ and $\mathbf{t}_{wc}$ is the rotation and translation which describes the transformation from the world coordinate frame, in which the 3D points M are measured, to the camera coordinate frame.

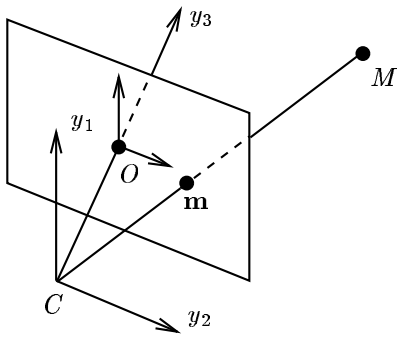

Fig. 1: The pinhole camera model

The geometry of the motion problem of line segments is illustrated in Fig. 2, where the measurements related to the second camera are identified by a prime $'$. The transformation from the coordinate frame associated to the first camera to that associated to the second is described by $(\mathbf{R}, \mathbf{t})$: the first camera undergoes a rotation $\mathbf{R}$ followed by a translation $\mathbf{t}$ in order to arrive at the position of the second camera.

An image line segment $\mathbf{l}$ is represented by its starting point $\mathbf{s}$ and its endpoint $\mathbf{e}$. We assume that line segments are oriented, which can be obtained through the information about the intensity contrast. As we can observe later, the orientation information is not indispensable in motion and structure problem. However, this information is in general available, and has usually already been used in the matching process.

We are given two line segments, $\mathbf{l}$ in camera 1 and $\mathbf{l}'$ in camera 2, in correspondence. The basic assumption we will use in this paper is that they are projections of two portions of a line segment SE in space and that the two portions *share a common part* (i.e. they overlap). We do not assume that the starting points ($\mathbf{s}$ and $\mathbf{s}'$) and the endpoints ($\mathbf{e}$ and $\mathbf{e}'$) are in correspondence. Indeed, these points are not reliable, mainly for three reasons:

- The first is purely algorithmic: because of noise in the images and because sometimes we approximate contours which are significantly curved with line segments, the polygonal approximation may vary from frame to frame, inducing a variation in the segments endpoints.
- The second is physical: because of partial occlusion in the scene, a segment can be considerably shortened or lengthened, and the occluded part may change over time.
- The third is photometric: because lighting and surface reflection often change when the view point changes, the segments endpoints may vary from frame to frame.

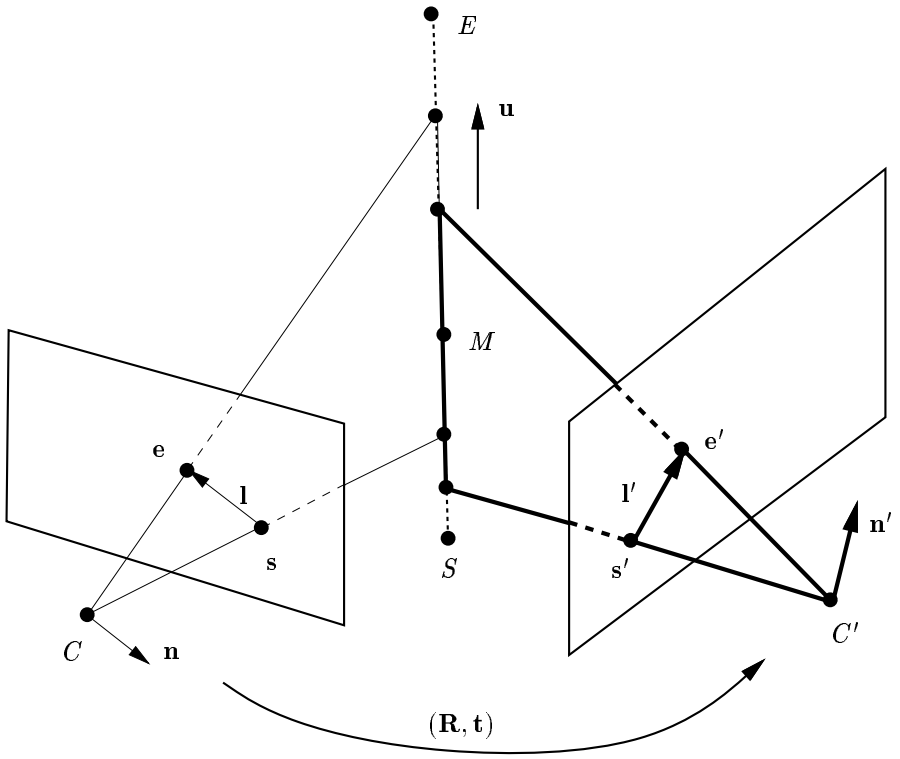

Fig. 2: The geometry of the motion of line segments

However, the location and orientation of a line segment can generally be reliably determined by fitting a line to a set of linked edge points [9].

Now, the problem to be solved in this paper can be stated as follows:

> **Given** two sets of line segments, $\{(\mathbf{l}_i, \mathbf{l}'_i) \mid i = 1, \ldots, n\}$, which are in correspondence,
> **Estimate** the camera motion parameters $(\mathbf{R}, \mathbf{t})$, and eventually determine the structure of the scene.

2.3 Projective lines and points in image plane

Let $\mathbf{n}$ be the normal vector of the plane which passes through the line segment $\mathbf{l}$ and the optical center C (which is sometimes called the *projection plane* of the line segment, see Fig. 2). The vector $\mathbf{n}$ actually defines the infinite line supporting the line segment. More precisely, $\mathbf{n}$ is the *projective representation* of the line $\mathbf{l}$ in image plane (without ambiguity, we use $\mathbf{l}$ to denote both the line segment and its supporting line). As the image plane is parallel to y_1y_2 plane in the coordinate frame associated to the first camera, the coordinate frame attached to the image plane is Oy_1y_2. For a point $\mathbf{a} = [u, v]^T$ in the image plane, its homogeneous coordinates are $\widetilde{\mathbf{a}} = [u, v, 1]^T$ corresponding to the same point in $Cy_1y_2y_3$ coordinate frame, and its projective coordinates will be $\lambda\widetilde{\mathbf{a}}$ for any nonzero scalar λ. For any point $\mathbf{m} = [u, v]^T$ on the infinite supporting line $\mathbf{l}$, we have the following relation:

$$\mathbf{n}^T\widetilde{\mathbf{m}} = 0 \, .$$

As one can observe in the above equation, points and lines play a symmetric role. This is known as the *principle of duality*. In the sequel, if there is no ambiguity, when we talk about an image line $\mathbf{l}$, the vector $\mathbf{l}$ is the projective representation $\mathbf{n}$ of the line, i.e. $\mathbf{l} = \mathbf{n}$.

Working with projective coordinates provides us with simple mathematical tools. In particular, we will need the following two elementary operations [6, chapter 2]: the line defined by two points $\mathbf{a}$ and $\mathbf{b}$ is represented by $\mathbf{l} = \widetilde{\mathbf{a}} \times \widetilde{\mathbf{b}}$; the intersection point of two lines $\mathbf{l}_1$ and $\mathbf{l}_2$ is represented by $\widetilde{\mathbf{m}} = \mathbf{l}_1 \times \mathbf{l}_2$. Dividing the first two elements of $\widetilde{\mathbf{m}}$ by the third element gives the Euclidean coordinates of the point in the image plane. Thus, the infinite line supporting the line segments defined by points $\mathbf{s}$ and $\mathbf{e}$ is represented by $\mathbf{n} = \widetilde{\mathbf{s}} \times \widetilde{\mathbf{e}}$.

2.4 Unlike supporting lines, line segments can determine the motion

Refer again to Fig. 2. Assume the world coordinate frame coincides with the coordinate frame of the first camera. Given two line segments $\mathbf{l}$ and $\mathbf{l}'$ in correspondence, let the

corresponding 3D line segment be represented by its direction vector $\mathbf{u}$ and a point M on it. Clearly, we have $\mathbf{n} = M \times \mathbf{u}$ for the first camera. Working in the coordinate frame of the second camera, we have

$$\mathbf{n}' = (\mathbf{R}M + \mathbf{t}) \times (\mathbf{R}\mathbf{u}) = \mathbf{R}(M \times \mathbf{u}) + \mathbf{t} \times (\mathbf{R}\mathbf{u}) ,$$

i.e.
$$\mathbf{n}' = \mathbf{R}\mathbf{n} + \mathbf{t} \times (\mathbf{R}\mathbf{u}) .$$

Multiplying the above equation by $\mathbf{t}$ to eliminate the unknown $\mathbf{u}$, we have

$$\mathbf{t}^T(\mathbf{n}' - \mathbf{R}\mathbf{n}) = 0 .$$

This equation is useless, however, because during its derivation, $\mathbf{n}$ and $\mathbf{n}'$ where related to a single point M on the 3D line, which is unknown. We have only a projective representation of the line, and so $\mathbf{n}$ and $\mathbf{n}'$ can be multiplied by an arbitrary scalar which is not equal to zero.

Indeed, it has been well known that motion cannot be determined from two views of straight lines. Geometrically, it is obvious: let us fix the position and orientation of the first camera. Now we move the second camera to another position and orientation. For each image line, its corresponding 3D line must lie on the plane (called the *projection plane* of the line) passing through the optical center and the image line. For each pair of lines in correspondence, we have a pair of projection planes, whose intersection determines the 3D line in space. The structure of the scene can so be determined. However, any two planes define a line. We can move the second camera to an arbitrary position and orientation, and we still obtain a 3D structure consistent with the two images. In other words, two sets of lines do not constrain the motion of the camera. If a third image is available, the motion and structure can in general be uniquely determined because three projection planes generally do not define a line.

When line segments are considered, the motion of the second camera can no longer be arbitrary. Indeed, each line segment defines a *generalized triangle* in space with the first side passing through the optical center C and the starting point $\mathbf{s}$ of the line segment, the second side passing through C and the endpoint $\mathbf{e}$, and the third side at infinity (see Fig. 3). Two such triangles generally do not intersect, and it is thus possible to determine the motion and structure of the scene from correspondences of line segments in two images.

3 Solving the Motion Problem by Maximizing the Overlap of Line Segments

In this section, we present the algorithm for solving the motion problem by maximizing the overlap of line segments. The epipolar constraint, as the base of the algorithm, is described first.

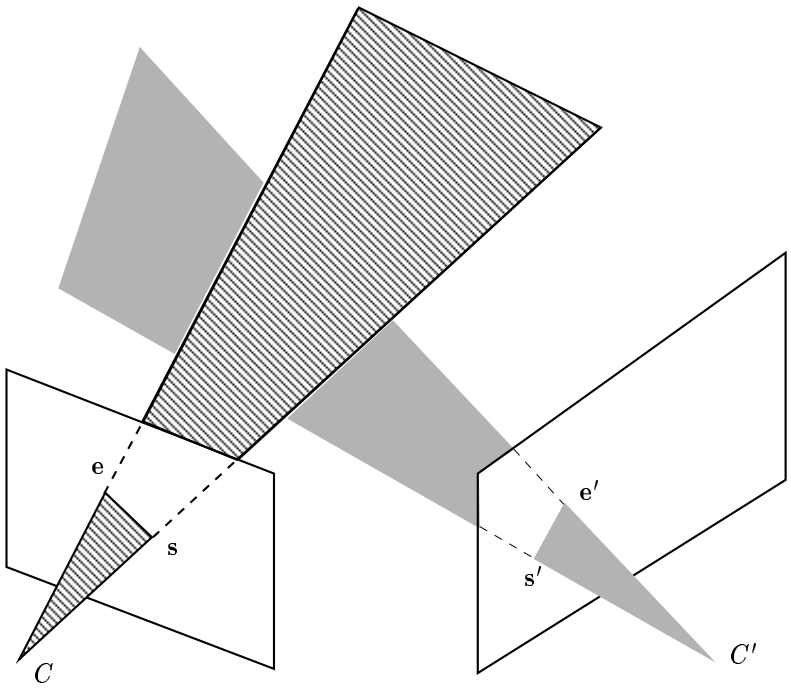

Fig. 3: Motion and structure from line segments

3.1 Epipolar constraint

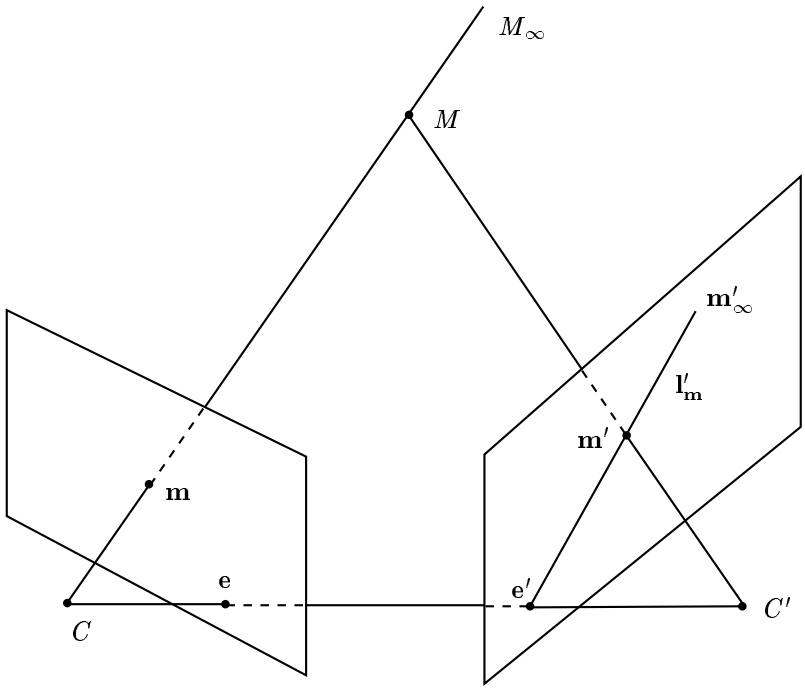

Fig. 4: Epipolar geometry

Refer to Fig. 4. Given a point $\mathbf{m}$ in the first image, its corresponding point M in space must be on the semi-line CM_∞ passing through $\mathbf{m}$, where M_∞ is a point at infinity. Letting the world coordinate frame coincide with the coordinate frame associated to the first camera, and letting $\mathbf{m} = [u, v]^T$, then point M can be represented as

$$M = \lambda \widetilde{\mathbf{m}} = \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad \lambda \in (0, \infty) .$$

This is in fact the parametric representation of the semi-line CM_∞. If we express this point in the coordinate frame of the second camera, we have

$$M' = \mathbf{R}M + \mathbf{t} = \lambda \mathbf{R}\widetilde{\mathbf{m}} + \mathbf{t} , \quad \lambda \in (0, \infty) .$$

The projection of the semi-line CM_∞ on the second camera is still a line, denoted by $\mathbf{l}'_\mathbf{m}$, on which the corresponding point in the second image of point $\mathbf{m}$ must lie. The line $\mathbf{l}'_\mathbf{m}$ is known as the *epipolar line* of $\mathbf{m}$. The above constraint is known as the *epipolar constraint*.

The epipolar line can be defined by two points. The first point can be obtained by projecting M' with $\lambda = 0$, which gives $\widetilde{\mathbf{e}}' = \frac{1}{t_3}\mathbf{t}$, where t_3 is the third element of the translation vector $\mathbf{t}$, or projectively, $\widetilde{\mathbf{e}}' = \mathbf{t}$. This is in fact the projection of the optical center C of the first camera on the second camera. The second point can be obtained by projecting M with $\lambda = \infty$, which gives $\widetilde{\mathbf{m}}'_\infty = \frac{1}{\mathbf{r}_3^T\widetilde{\mathbf{m}}}\mathbf{R}\widetilde{\mathbf{m}}$, where $\mathbf{r}_3$ is the third row of the rotation matrix $\mathbf{R}$, or projectively $\widetilde{\mathbf{m}}'_\infty = \mathbf{R}\widetilde{\mathbf{m}}$. As described in Sect. 2.3, the epipolar line $\mathbf{l}'_\mathbf{m}$ is projectively represented by

$$\mathbf{l}'_\mathbf{m} = \widetilde{\mathbf{e}}' \times \widetilde{\mathbf{m}}'_\infty = \mathbf{t} \times \mathbf{R}\widetilde{\mathbf{m}} \,. \tag{2}$$

We introduce the antisymmetric matrix $[\mathbf{t}]_\times$:

$$[\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} ,$$

defined by a vector $\mathbf{t} = [t_1, t_2, t_3]^T$. The matrix $[\mathbf{t}]_\times$ is such that $[\mathbf{t}]_\times\mathbf{x} = \mathbf{t} \times \mathbf{x}$ for all vector $\mathbf{x}$. Letting $\mathbf{E} = [\mathbf{t}]_\times\mathbf{R}$, Equation 2 can be rewritten as

$$\mathbf{l}'_\mathbf{m} = \mathbf{E}\widetilde{\mathbf{m}} \,. \tag{3}$$

The matrix $\mathbf{E}$ is the well-known *essential matrix* [13].

It is easy to see that all epipolar lines in the second image pass through the single point $\mathbf{e}'$. Indeed, the epipolar lines in the second image are the projections of the pencil of semi-lines all starting at the optical center C of the first camera, and they necessarily go through the point $\mathbf{e}'$ which is the projection of C. The point $\mathbf{e}'$ is thus called the *epipole* in the second image.

If now we reverse the role of the two camera, we find the epipolar geometry is symmetric for the two cameras. Indeed, the epipole $\mathbf{e}$ in the first image is the projection of the optical center of the second camera, which is given by $\widetilde{\mathbf{e}} = -\mathbf{R}^T\mathbf{t}$. For a given point $\mathbf{m}'$ in the second image, its corresponding epipolar line in the first image is

$$\mathbf{l}_{\mathbf{m}'} = -(\mathbf{R}^T\mathbf{t}) \times (\mathbf{R}^T\widetilde{\mathbf{m}}') = -\mathbf{R}^T[\mathbf{t}]_\times\widetilde{\mathbf{m}}' = \mathbf{E}^T\widetilde{\mathbf{m}}' \,.$$

It is seen that the transpose of matrix $\mathbf{E}$, $\mathbf{E}^T$, defines the epipolar lines in the first image.

3.2 Overlap of two corresponding line segments

Let us consider the situation illustrated in Fig. 5. We are given a pair of line segments $(\mathbf{l}, \mathbf{l}')$ in correspondence. The line $\mathbf{l}'_\mathbf{s}$ in the second image is the epipolar line of $\mathbf{s}$, i.e. $\mathbf{l}'_\mathbf{s} = \mathbf{E}\widetilde{\mathbf{s}}$; the

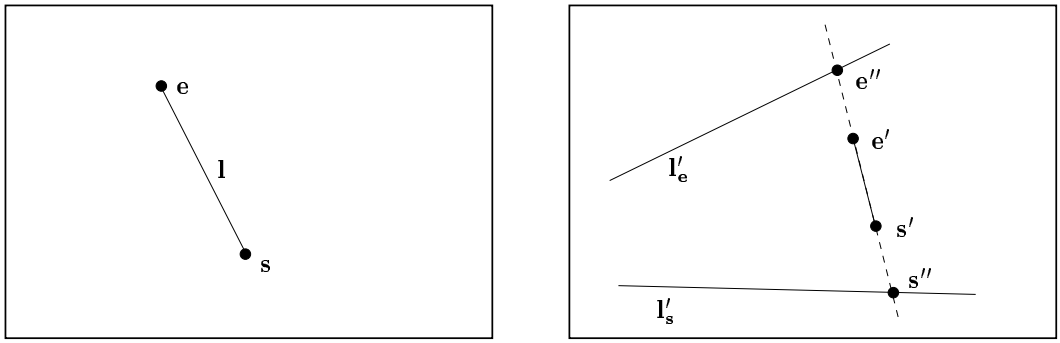

Fig. 5: Overlap of two line segments in correspondence

line $\mathbf{l}'_{\mathbf{e}}$ is the epipolar line of $\mathbf{e}$, i.e. $\mathbf{l}'_{\mathbf{e}} = \mathbf{E}\widetilde{\mathbf{e}}$. We denote the intersection of $\mathbf{l}'_{\mathbf{s}}$ with line $\mathbf{l}'$ by $\widetilde{\mathbf{s}}'' = \mathbf{l}' \times \mathbf{l}'_{\mathbf{s}}$, and the intersection of $\mathbf{l}'_{\mathbf{e}}$ with line $\mathbf{l}'$ by $\widetilde{\mathbf{e}}'' = \mathbf{l}' \times \mathbf{l}'_{\mathbf{e}}$.

Provided that the epipolar geometry (i.e. matrix $\mathbf{E}$, or the motion $(\mathbf{R}, \mathbf{t})$) between two images is correct, then $\mathbf{s}$ and $\mathbf{s}''$ correspond to a single point in space; so do $\mathbf{e}$ and $\mathbf{e}''$. Thus, the statement that two line segments $\mathbf{l}$ and $\mathbf{l}'$ share a common part of a 3D line segment is equivalent to saying that line segment $\mathbf{s}''\mathbf{e}''$ and line segment $\mathbf{s}'\mathbf{e}'$ (i.e. $\mathbf{l}'$) overlap. In order for $\mathbf{s}'\mathbf{e}'$ and $\mathbf{s}''\mathbf{e}''$ to overlap, the following two conditions must be satisfied:

1. $\mathbf{e}'$ and $\mathbf{e}''$ must be on the same side with respect to $\mathbf{s}'$. This implies:

$$(\mathbf{e}'' - \mathbf{s}') \cdot (\mathbf{e}' - \mathbf{s}') > 0 \, .$$

2. $\mathbf{s}'$ and $\mathbf{s}''$ must be on the same side with respect to $\mathbf{e}'$. This implies:

$$(\mathbf{e}' - \mathbf{s}'') \cdot (\mathbf{e}' - \mathbf{s}') > 0 \, .$$

The above two constraints do not use the fact that the line segments are oriented. The configuration in Fig. 6 satisfies the above two conditions, but does not satisfy the orientation congruence. In order to assure the orientation congruence, we must impose another constraint:

3. Line segments $\mathbf{s}'\mathbf{e}'$ and $\mathbf{s}''\mathbf{e}''$ should be oriented in the same way. This implies:

$$(\mathbf{e}' - \mathbf{s}') \cdot (\mathbf{e}'' - \mathbf{s}'') > 0 \, .$$

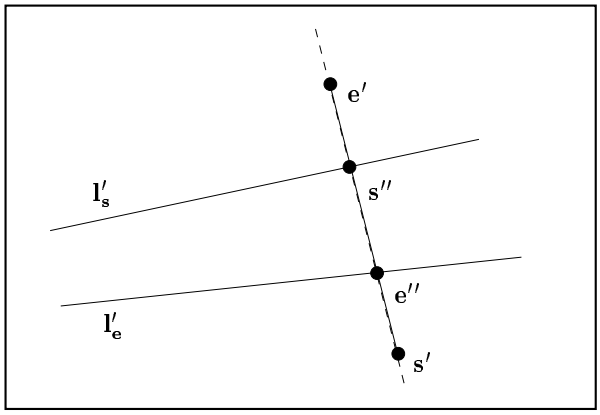

Fig. 6: Two incongruent line segments in orientation

Note that it is here that the orientation information of a line segment is used. Remove this constraint, and the proposed algorithm will work for line segments which are not oriented.

Thus, the problem of motion and structure from correspondences of line segments can be solved by nonlinear programming such that the above three inequality are satisfied for each correspondence. Our experience shows, however, that all motion parameters near the true solution are feasible, making the motion estimation not very precise. In the next, we solve the problem by maximizing the overlap of line segments.

3.3 Estimating the motion by maximizing the overlap

We first define a measure of overlap, which we will call the *overlap length*, for two line segments in correspondence. The overlap length is positive if two line segments overlap; otherwise, it is negative.

If the conditions described in last subsection are satisfied, the two line segments overlap, and we can easily see that there exist only four configurations of overlap as illustrated in Fig. 7. The overlap length, denoted by $\mathcal{L}'$, is defined as the length of the common segment, which is given by

$$\mathcal{L}' = \min(\|\mathbf{e}' - \mathbf{s}'\|, \|\mathbf{e}'' - \mathbf{s}'\|, \|\mathbf{e}' - \mathbf{s}''\|, \|\mathbf{e}'' - \mathbf{s}''\|) \ . \tag{4}$$

If two segments do not overlap, there are eight possible configurations, as shown in Fig. 8. The first two configurations are due to the normal separation of two segments, and we define

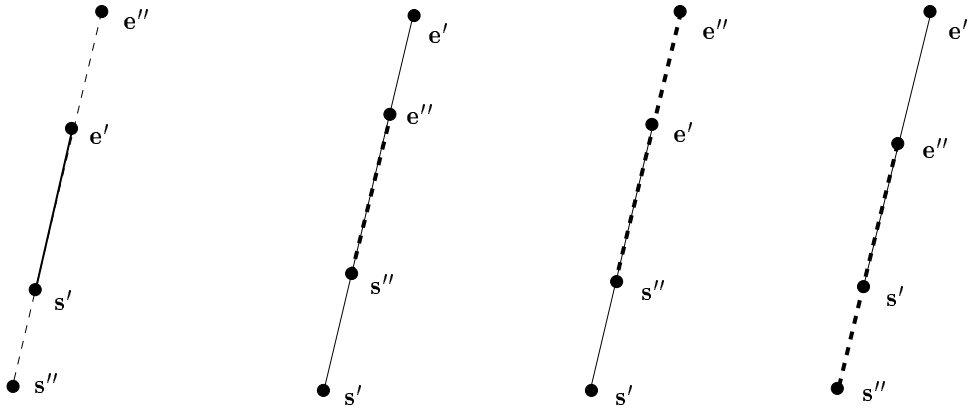

Fig. 7: Four configurations of two line segments with overlap

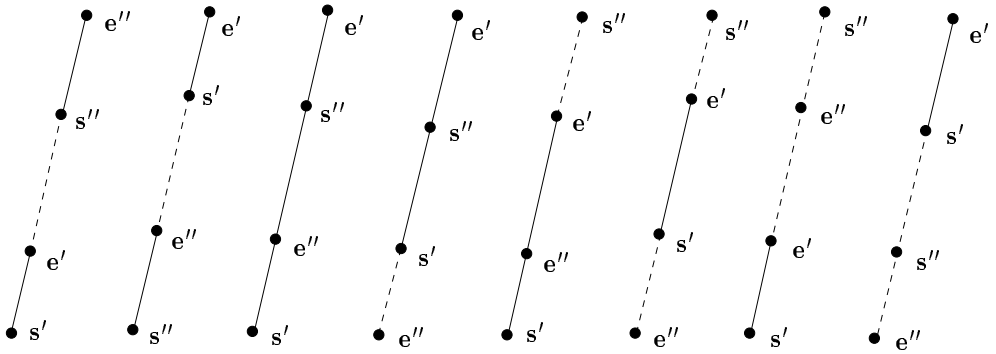

Fig. 8: Configurations of two line segments which do not overlap

the overlap length as

$$\mathcal{L}' = -\min(\|\mathbf{e}' - \mathbf{s}''\|, \|\mathbf{e}'' - \mathbf{s}'\|) \ , \tag{5}$$

which corresponds to the gap between the two segments. The other configurations are due to the inconsistence of line segment orientations. The overlap length for their configurations is defined as

$$\mathcal{L}' = -\max(\|\mathbf{s}' - \mathbf{s}''\|, \|\mathbf{e}' - \mathbf{e}''\|) \ . \tag{6}$$

The idea is to reduce the distance between the starting points and that between the end-points, and eventually to achieve the orientation congruence.

The above overlap measure of a given pair of line segments is defined in the second image. We have no reason for one image to prevail over another. In order for the two images to play a symmetric role, we can compute the overlap length in the first image, denoted by $\mathcal{L}$, exactly in the same way.

Since a small overlap length for a short line segment is as important as a large overlap length for a long line segment, it is more reasonable to use the *relative* overlap length, and thus we should use $\mathcal{L}_i/l_i$ and $\mathcal{L}'_i/l'_i$ to measure the overlap of a pair of line segments $(\mathbf{l}_i, \mathbf{l}'_i)$, where l_i and l'_i are the length of the line segments $\mathbf{l}_i$ and $\mathbf{l}'_i$, respectively. The relative overlap length takes a value between 0 and 1 when two segments overlap; otherwise it will be negative. Now we can formulate the motion problem as follows: Given n correspondences of line segments, $\{(\mathbf{l}_i, \mathbf{l}'_i) \mid i = 1, \ldots, n\}$, estimate the camera motion parameters $(\mathbf{R}, \mathbf{t})$ by minimizing the following objective function

$$\mathcal{F} = \sum_{i=1}^{n} \left((1 - \mathcal{L}_i/l_i)^2 + (1 - \mathcal{L}'_i/l'_i)^2 \right) \ . \tag{7}$$

3.4 Implementation details

The minimization of the objective function (7) is conducted using a *simplex* algorithm.

The rotation $\mathbf{R}$ is represented by a 3D vector $\mathbf{r} = [r_1, r_2, r_3]^T$, whose direction is that of the rotation axis and whose norm is the equal to the rotation angle. The vector $\mathbf{r}$ is related to the matrix $\mathbf{R}$ by the Rodrigues' formula [21]:

$$\mathbf{R} = \mathbf{I}_3 + \frac{\sin\theta}{\theta}[\mathbf{r}]_\times + \frac{1 - \cos\theta}{\theta^2}[\mathbf{r}]_\times^2 \ ,$$

where $\mathbf{I}_3$ is the 3×3 identity matrix, and $\theta = \|\mathbf{r}\|$.

The translation $\mathbf{t}$ is represented by its spherical coordinates (ϕ, θ), because the magnitude of $\mathbf{t}$ is inherently unrecoverable.

As the problem is nonlinear, an initial guess of the motion is required. We have tried to estimate the motion by assuming the correspondences of midpoints, but the results are useless. For the solution that works best, we choose to sample the parameter space to obtain a global minimum. For each component of the rotation axis, we assume it ranges from $-(\frac{\pi}{4}+\alpha)$ to $(\frac{\pi}{4}+\alpha)$, where α is an increment on the search bound we use such that the algorithm still converges. This range is sufficient for most applications of motion analysis. We then sample the range $[-\frac{\pi}{4}, \frac{\pi}{4}]$ with step equal to $\frac{\pi}{8}$. We have thus $5^3 = 125$ samples of rotation.

The samples of translation are obtained through a uniform partition of a Gauss sphere based on the icosahedron [2]. The icosahedron has 12 vertices, 20 faces and 30 edges. Basically, we obtain 20 samples of 3D directions. Adding its dual (vertices) yields in total 32 samples. To obtain more samples, we further divide each icosahedral edge into n equal lengths and construct n^2 congruent equilateral triangles on each face, pushing them out to the radius of the sphere for their final position. In particular, for $n = 2$, we have 80 samples; for $n = 3$, we have 180 samples. From our experience, we found that 80 samples are sufficient for solving the problem in hand.

As a matter of fact, we do not need to use all 80 samples because of the following proposition. We only need half of them, i.e. the samples from a semi-sphere.

Proposition 1 *We consider given two sets of points $\{(\mathbf{m}_i, \mathbf{m}'_i)\}$ (the same for line segments) in correspondence. If $(\mathbf{R}, \mathbf{t}, \{M_i\})$ is a solution of the motion and structure, then $(\mathbf{R}, -\mathbf{t}, \{-M_i\})$ is also a solution.*

Proof: Under the pinhole model, we have

$$\begin{cases} s_i\widetilde{\mathbf{m}}_i = & [\mathbf{I}\ \mathbf{0}]\widetilde{M}_i & = M_i & \text{(for the first image)} \\ s'_i\widetilde{\mathbf{m}}'_i = & [\mathbf{R}\ \mathbf{t}]\widetilde{M}_i & = \mathbf{R}M_i + \mathbf{t} & \text{(for the second image)} \end{cases}$$

where s_i and s'_i are arbitrary scalars. It is evident that if $(\mathbf{R}, \mathbf{t}, \{M_i\})$ is a solution to the motion and structure problem, then $(\mathbf{R}, -\mathbf{t}, \{-M_i\})$ is also a solution. This is because if s_i and s'_i are the scale factors for the first solution, we obtain the second solution with scale factors $-s_i$ and $-s'_i$. Both solutions are compatible with the observed data. ∎

It is thus inherently impossible to determine geometrically the sign of the translation vector from two perspective images. So we only need to sample a semi-sphere for the translation. The ambiguity can be resolved by imposing some physical constraint, e.g. the reconstructed points should be in front of the cameras (i.e. they have positive depth). If their depths are negative, it is sufficient, from the above proposition, to multiply $\mathbf{t}$ and $\{M_i\}$ by -1 to obtain the physical solution.

In passing, if we do not impose that matrix $\mathbf{R}$ is a rotation matrix, then $(-\mathbf{R}, \mathbf{t}, \{-M_i\})$ and $(-\mathbf{R}, -\mathbf{t}, \{M_i\})$ are two other solutions. However, if the original solution $\mathbf{R}$ is a rotation, i.e. $\det \mathbf{R} = 1$, then these two solutions correspond to a reflection because $\det(-\mathbf{R}) = -1$.

To summarize, we have $125 \times 40 = 5000$ sample points in the motion space. We evaluate the objective function for each sample, and retain 10 samples which yield the smallest values of the objective function. All 10 samples are used as the initial guess to carry out the minimization procedure independently. At the end, the one which produces the smallest value of the objective function is considered as the solution of the motion. To give an idea of the time complexity, it takes about 4.3 seconds on a SPARC 10 station to perform a complete run of the algorithm for 35 line segments correspondences.

4 Reconstructing 3D Line Segments

Once we have an estimate of the motion between two images, we can reconstruct the 3D line segment for each pair of image line segments $(\mathbf{l}, \mathbf{l}')$ in correspondence.

We first compute the infinite 3D line, which is the intersection of the two projection planes. The line can be represented by its direction vector $\mathbf{u}$ and a point $\mathbf{x}$ on it, say, the point which is closest to the origin of the coordinate frame of the first camera. For the direction vector $\mathbf{u}$, we have

$$\begin{cases} \mathbf{n}^T\mathbf{u} = 0 & (\mathbf{u} \text{ is in the first projection plane}) \\ \mathbf{n}'^T(\mathbf{R}\mathbf{u}) = 0 & (\mathbf{u} \text{ is in the second projection plane}) \end{cases}$$

which gives

$$\mathbf{u} = \mathbf{n} \times (\mathbf{R}^T\mathbf{n}') \, .$$

For the point $\mathbf{x}$, we have

$$\begin{cases} \mathbf{n}^T\mathbf{x} = 0 & (\mathbf{x} \text{ is in the first projection plane}) \\ \mathbf{n}'^T(\mathbf{R}\mathbf{x} + \mathbf{t}) = 0 & (\mathbf{x} \text{ is in the second projection plane}) \\ \mathbf{u}^T\mathbf{x} = 0 & (\mathbf{x} \text{ is closest to } C) \end{cases}$$

The solution is:

$$\mathbf{x} = \begin{bmatrix} \mathbf{n} & \mathbf{R}^T\mathbf{n}' & \mathbf{n} \times (\mathbf{R}^T\mathbf{n}') \end{bmatrix}^{-T} \begin{bmatrix} 0 \\ -\mathbf{t}^T\mathbf{n}' \\ 0 \end{bmatrix} \, .$$

It is then trivial to recover the point on the 3D line corresponding to each endpoint of the image line segments, and we have two 3D line segments. It remains the choice of the appropriate 3D line segments. Due to the reasons described in Sect. 2.2, a 2D line segments is only an observation of a portion of the real line segment in space. Two segments are considered to be matched if they have a common part. Their corresponding segment in space can be expected not to be shorter than either of the two segments. That is, the *union* of the two segments can be reasonably considered as a better estimate of the corresponding segment in space.

5 Experimental Results

The proposed algorithm has been tested with both synthetic and real data. The error of the motion estimate is quantified by three numbers: the difference between the true and estimated rotation angles (indicated as `Dif. in Rot. Angle` in the figures), the angle between the true and estimated rotation axes (indicated as `Dif. in Rot. Axis` in the figures), and the angle between the true and estimated translation vectors (indicated as `Dif. in Trans. Vector` in the figures).

5.1 Synthetic Data

The synthetic data includes two objects: a box and a model car. The box has 15 line segments and the model car has 17 segments. A *realistic* stereo set-up is used. "Realistic" means the intrinsic and extrinsic parameters are those of a real stereo system used in our laboratory. The noise-free 3D line segments are then projected on each camera, and we obtain two sets of 2D noise-free line segments, whose rotation and translation is precisely known. To give an idea of the synthetic data, Fig. 9 shows the projection of the line segments on the first camera (perspective view) and on the ground plane (top view).

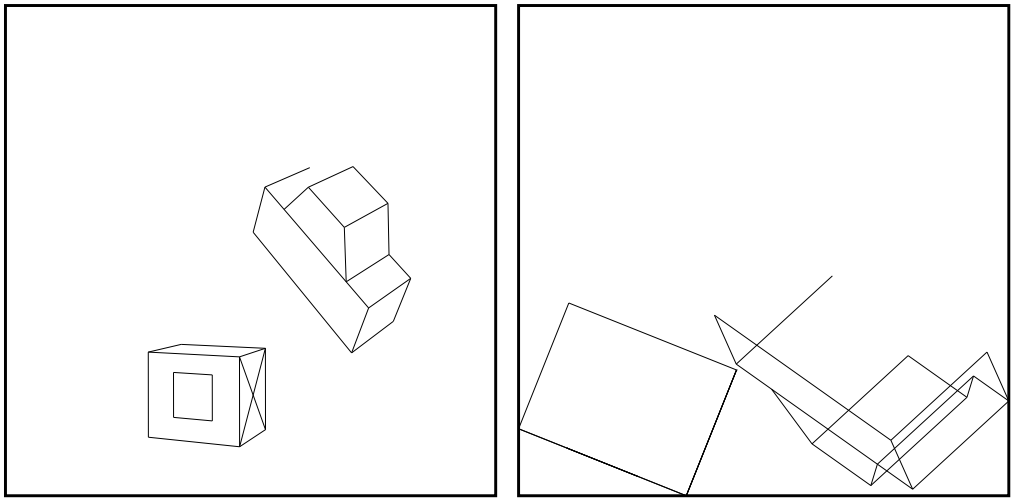

Fig. 9: Noise-free synthetic data: Perspective and top views

We add Gaussian noise to each endpoint of each projected line segment. The noise for each endpoint of a projected segment (see Fig. 10) has two independent components: one component parallel to the segment, denoted by $\sigma_{\parallel}$, and another perpendicular to the segment, denoted by $\sigma_{\perp}$. The perpendicular component is a random scalar with mean zero

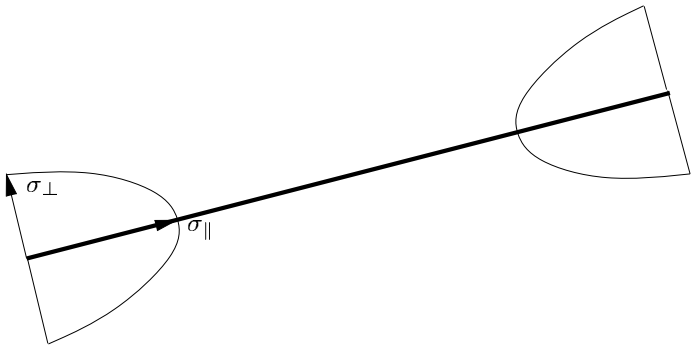

Fig. 10: Noise model used in synthetic data generation

and standard deviation $\sigma_\perp = 0.5$ pixels. The parallel component is a random scalar with mean zero and standard deviation $\sigma_\parallel$ variable with the length of the line segment. One detail is that we model the parallel component by a *half* Gaussian such that the added noise always *shortens* the line segment. We set $\sigma_\parallel = wl$, where l is the length of the segment, and w indicates the noise level. When w is small, two line segments in correspondence are expected to have important overlap; on the other hand, when w is large, there may be no overlap between two line segments.

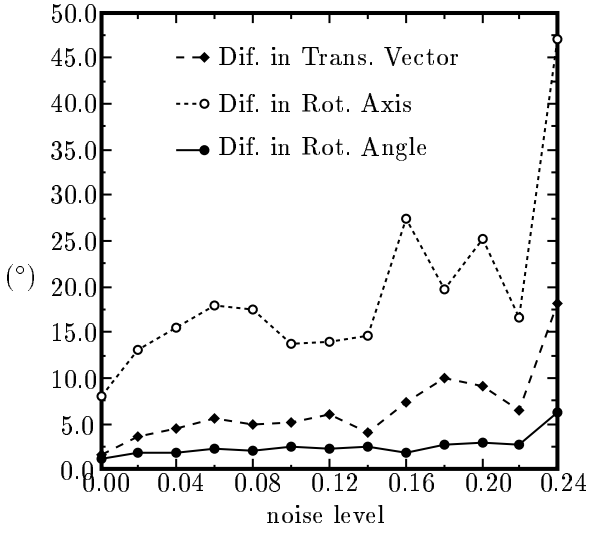

Fig. 11: Motion errors versus different noise levels: Data `Car`, 17 segments

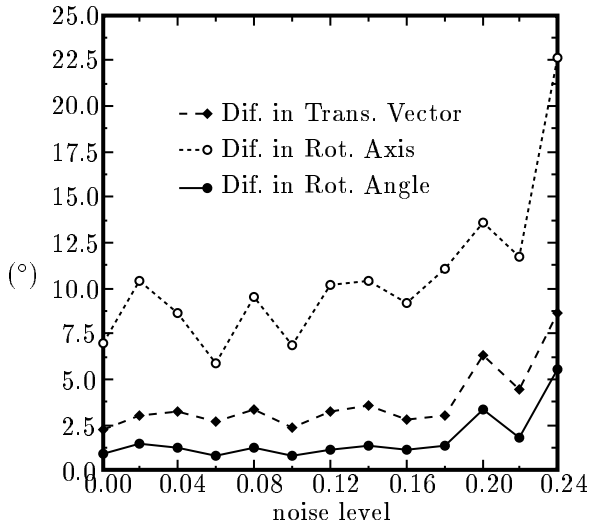

Fig. 12: Motion errors versus different noise levels: Data `All`, 32 segments

The first series of experiments we have carried out is on the sensibility of the proposed algorithm with respect to different noise levels w, namely different degrees of overlap between matched segments. We have tried the algorithm on the object `Car` alone (see Fig. 11), as well as on the whole set of data (see Fig. 12). We varied w from 0.0 to 0.24 with step 0.02. Note that even with $w = 0.0$ there is noise added to the endpoints because $\sigma_\perp$ is always equal to 0.5 pixels. For each noise level, 30 tries were performed, and the results shown in these figures are the average. The initial estimate is the true motion parameters. Several observations can be made from Fig. 11 and Fig. 12:

1. The errors are generally small with small value of w, but there is no prominent relationship between the errors and the noise level, except when $w \geq 0.2$. This implies that the proposed algorithm performs similarly well with different fragments of line segments as long as there is a reasonable overlap.

2. The errors decrease significantly with the increase of the number of line segments. The number of line segments in Data `All` is almost double of that in Data `Car`, and the errors with Data `All` are almost 50% less than those with Data `Car`.

The above two features are really what we expect.

The second series of experiments is on the sensibility of the algorithm with respect to different estimates of rotation while retaining the true translation estimate. We displace the

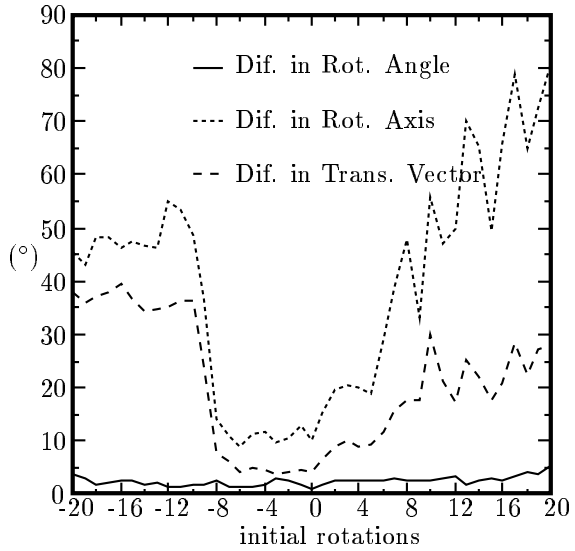

Fig. 13: Motion errors versus different initial rotations: $\mathbf{r}_{\text{true}} + i * 0.02 * [0,1,0]^T$

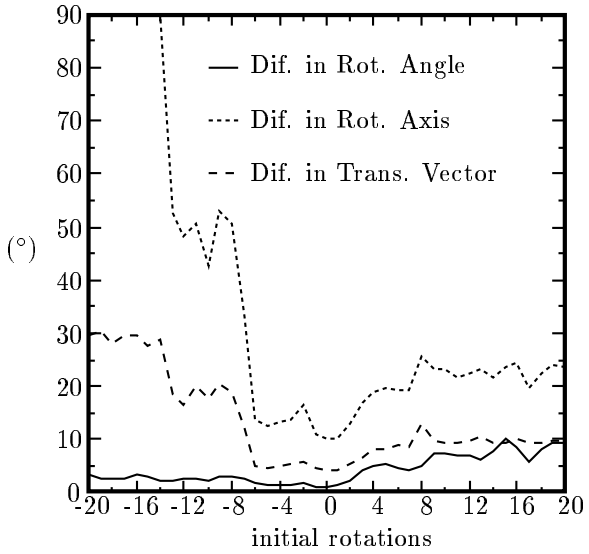

Fig. 14: Motion errors versus different initial rotations: $\mathbf{r}_{\text{true}} + i * 0.02 * [1,0,0]^T$

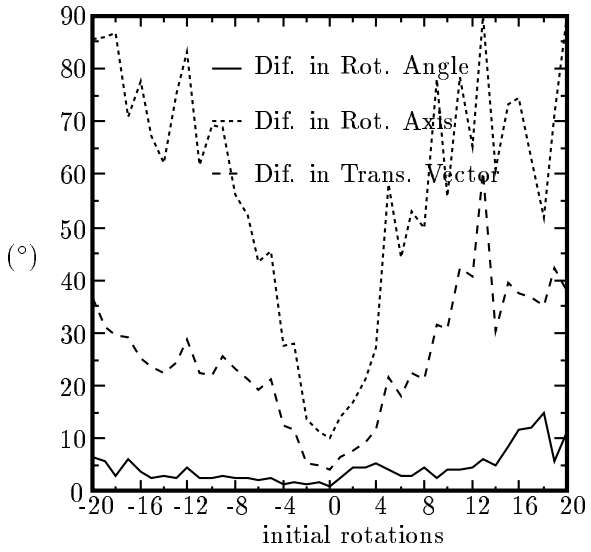

Fig. 15: Motion errors versus different initial rotations: $\mathbf{r}_{\text{true}} + i * 0.02 * [1,1,1]^T$

rotation vector $\mathbf{r}$ according to the following equation:

$$\mathbf{r} = \mathbf{r}_{\text{true}} + i * 0.02 * \mathbf{v} \ ,$$

where i varies from -20 to 20 with step equal to 1, and $\mathbf{v}$ is an arbitrary vector. The data set `All` with $w = 0.1$ was used, and 30 tries were generated. The results shown are the average of the 30 tries. Figures 13, 14, and 15 show the results corresponding to $\mathbf{v}$ equal to $[0,1,0]^T$, $[1,0,0]^T$, and $[1,1,1]^T$, respectively. From these results, we conclude that the initial rotation estimate has a strong influence in the final motion estimate. This is because the objective function is rather flat near the true solution when the number of line segments is large enough.

The third series of experiments is on the sensibility of the algorithm with respect to different estimates of translation while retaining the true rotation estimate. The same set of data is used as in the second series of experiments. We have rotated the true translation vector by $i * 0.05 * \mathbf{v}$ to obtain the initial estimate of the translation vector, where i varies from -20 to 20 with step equal to 1, and $\mathbf{v}$ is an arbitrary vector. Figures 16 and 17 show the results corresponding to $\mathbf{v}$ equal to $[0,1,0]^T$, and $[1,0,0]^T$, respectively. We observe that the final motion estimate is much less sensitive to different initial translation estimates than to different initial rotation estimates.

Finally, we provide an example through the sampling of the motion space as described in Sect. 3.4. The two sets of line segments used are shown in Fig. 18, corresponding to one try of

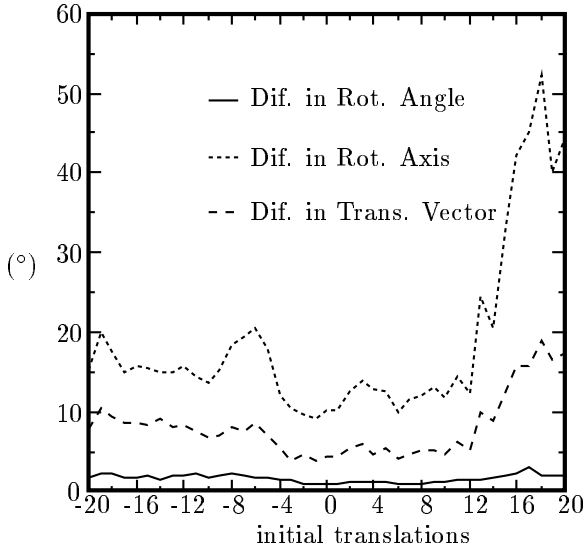

Fig. 16: Motion errors versus different initial translation: true translation rotated by $i * 0.05 * [0, 1, 0]^T$

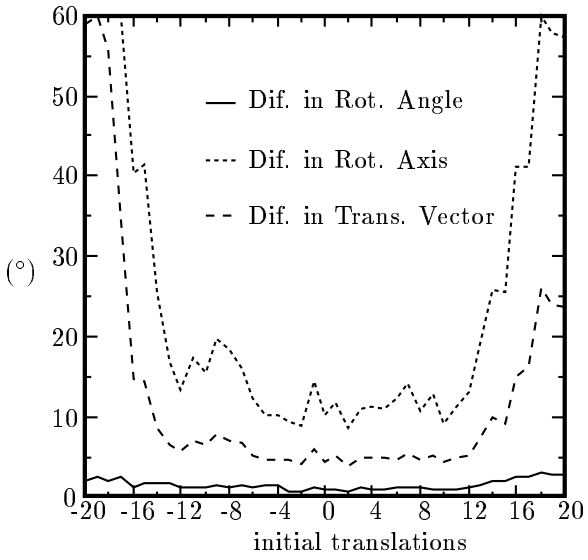

Fig. 17: Motion errors versus different initial translation: true translation rotated by $i * 0.05 * [1, 0, 0]^T$

data with $w = 0.1$. The difference between the true and estimated rotation angles is equal

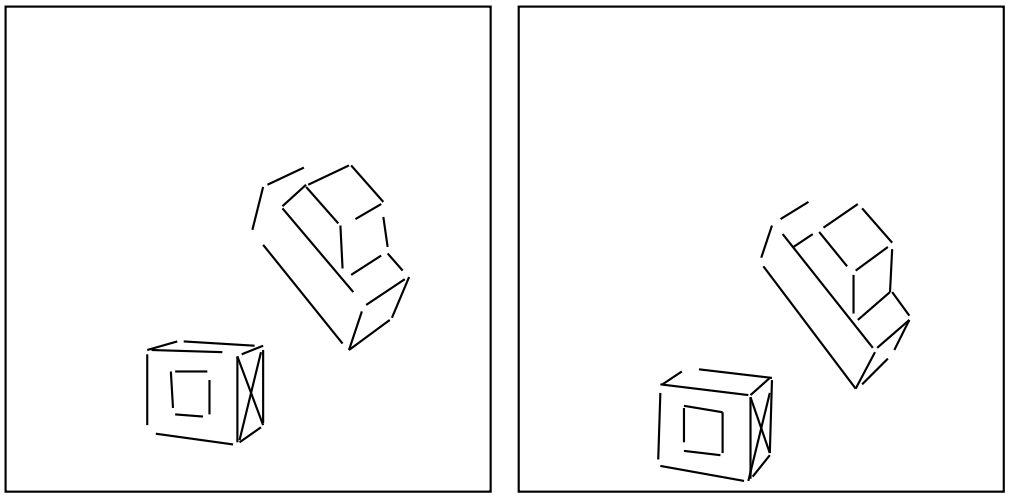

Fig. 18: Synthetic data: two sets of 2D line segments

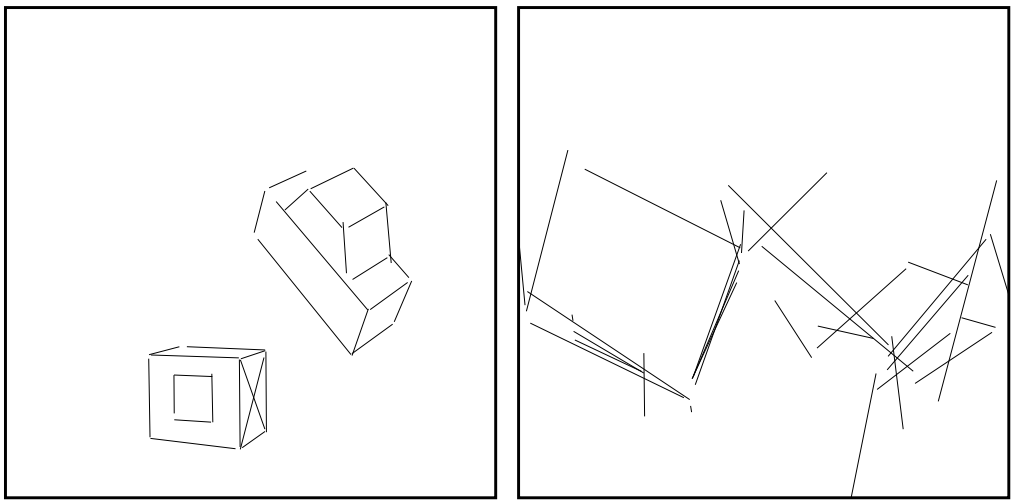

Fig. 19: 3D reconstruction of the synthetic data shown in Fig. 18: perspective and top views

to 1.21°; the angle between the true and estimated rotation axes is equal to 4.13°; the angle between the true and estimated translation vectors is equal to 4.7°. The 3D reconstruction is shown in Fig. 19, which should be compared with the original data shown in Fig. 9.

5.2 Real Data

We have tested our algorithm with three sets of real data which were extracted from a trinocular stereo system [1]. We have chosen the stereo data because the stereo system has been calibrated which serves as a ground truth [7].

The first set of real data is an image pair of an **Office** scene (see Fig. 20). 98 line segments have been matched by our trinocular stereo system, as shown in Fig. 21. One can notice quite a few false matches, such as the line segment near the top border, and one of the line segments on the table.

Fig. 20: Image pair of an **Office** scene

Through searching for the initial motion estimation by sampling as described in Sect. 3.4, the ten best samples all converge to the good solution. The motion estimation given by the algorithm described in this paper is:

$$\mathbf{r} = [1.424e{-}1, 5.510e{-}2, 5.388e{-}2]^T \ , \ \mathbf{t} = [-4.276e{-}1, 9.019e{-}1, 6.114e{-}2]^T \ ,$$

while the estimation through stereo calibration is:

$$\mathbf{r} = [1.468e{-}1, 5.736e{-}2, 6.254e{-}2]^T \ , \ \mathbf{t} = [-4.276e{-}1, 8.933e{-}1, 1.386e{-}1]^T \ .$$

The difference in the rotation angle is $0.436°$; the angle between the rotation axes is $2.217°$; the angle between the translation vectors is $4.469°$.

The 3D reconstruction based on the technique described in Sect. 4 is displayed in Fig. 22, where the picture on the left is the perspective view from the first camera and the one on

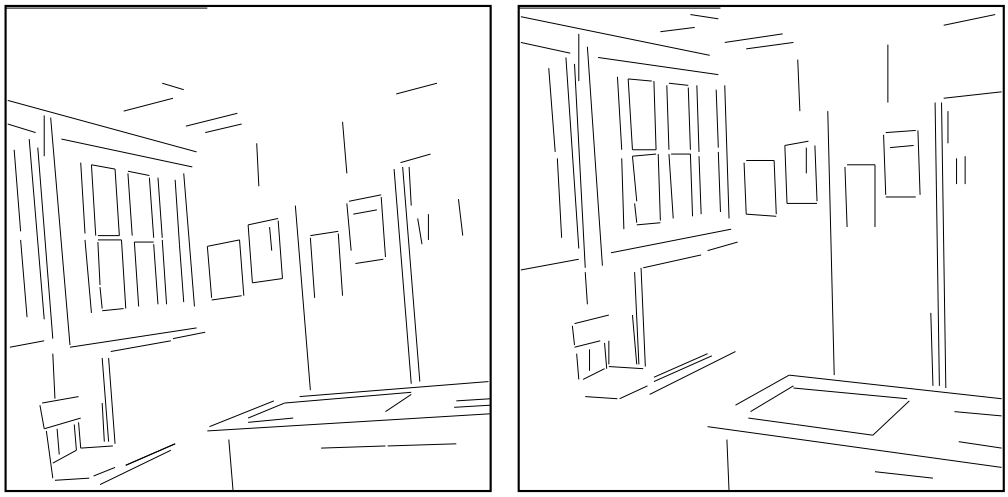

Fig. 21: Matched line segments of the **Office** image pair

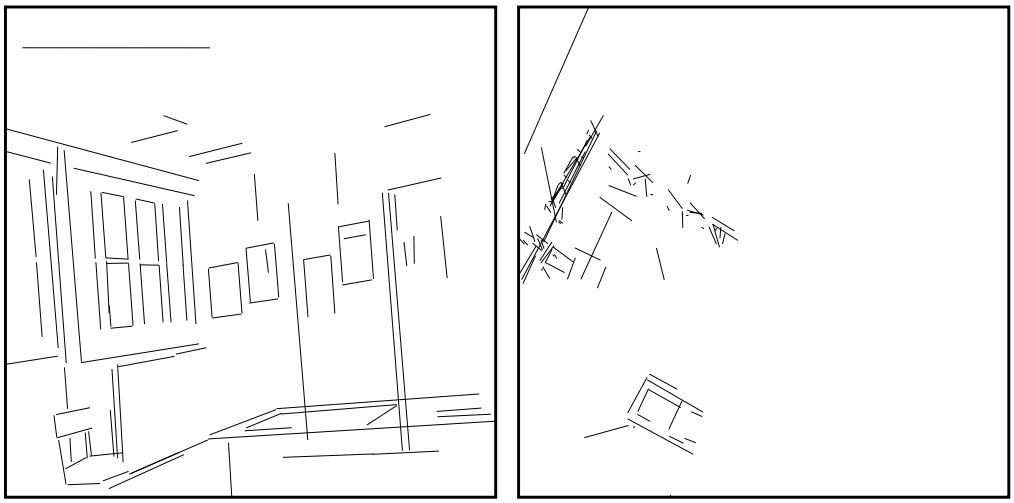

Fig. 22: 3D reconstruction of the **Office** scene by the structure from motion technique described in this paper: back projection on the first camera and projection on the ground plane

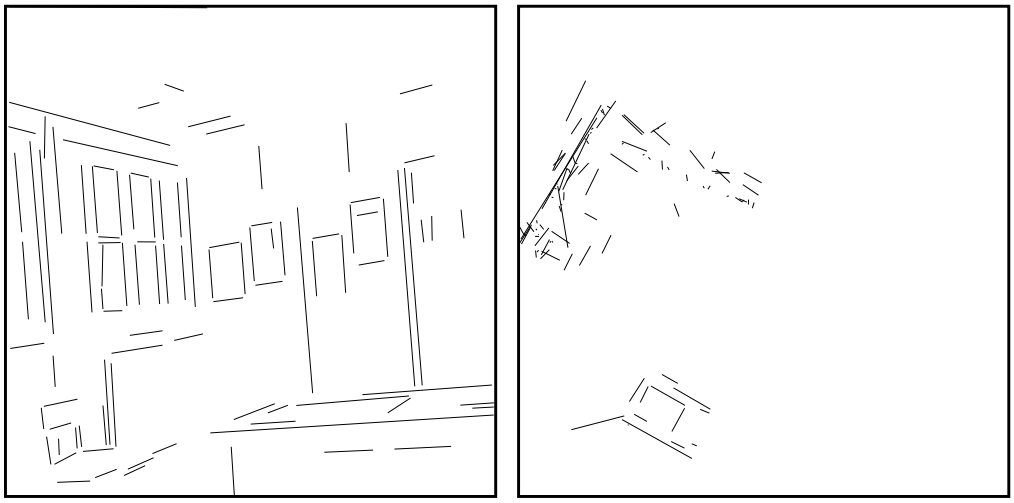

Fig. 23: 3D reconstruction of the **Office** scene by a classical trinocular stereo: back projection on the first camera and projection on the ground plane

the right is a top view. This result should be compared with that reconstructed by our trinocular stereo which uses *three* images (we used only two for the other algorithm) and whose geometry has been previously calibrated (see Fig. 23). Essentially, the same result can be observed. Because of use of the *union* strategy described in Sect. 4, the 3D reconstruction shown in Fig. 22 appears more complete than that shown in Fig. 23.

The second set of real data is an image pair of a scene named **Room** (see Fig. 24). 90 line segments have been matched by our trinocular stereo system, as shown in Fig. 25. One can easily notice two false matches located at the lower right corner near the border of the table.

The same algorithm has been applied to this set of data. Probably because of the gross error made in matching, only six of the ten best samples converge to the good solution. The final motion estimation is:

$$\mathbf{r} = [9.968e{-}2, 1.732e{-}1, 1.193e{-}2]^T \ , \ \mathbf{t} = [-8.186e{-}1, 5.237e{-}1, -2.357e{-}1]^T \ ,$$

while the estimation through stereo calibration is:

$$\mathbf{r} = [9.965e{-}2, 1.584e{-}1, 2.226e{-}2]^T \ , \ \mathbf{t} = [-7.768e{-}1, 6.182e{-}1, -1.198e{-}1]^T \ .$$

The difference in the rotation angle is $0.673°$; the angle between the rotation axes is $4.048°$; the angle between the translation vectors is $8.905°$.

The 3D line segments reconstructed by our algorithm are shown in Fig. 26, while those reconstructed by the trinocular stereo are shown in Fig. 27. The reconstruction correspon-

Fig. 24: Image pair of a **Room** scene

Fig. 25: Matched line segments of the **Room** image pair

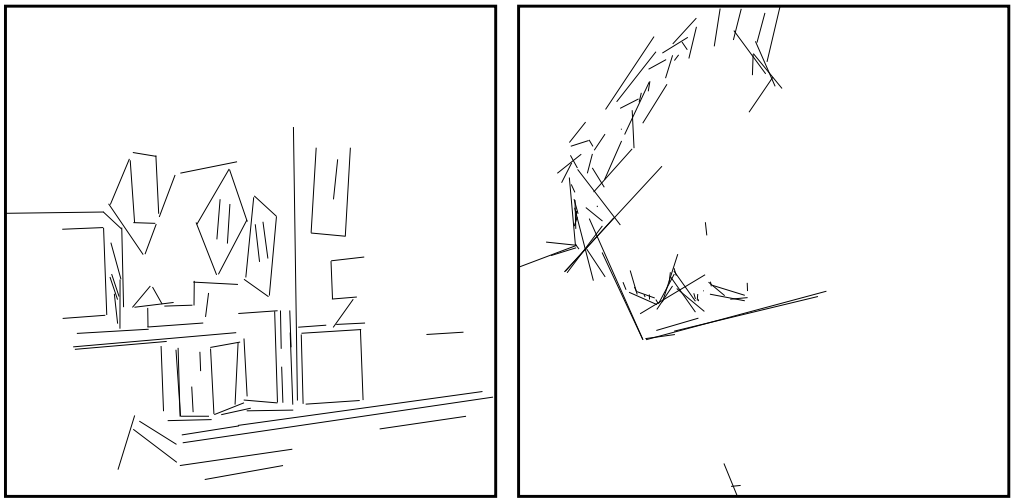

Fig. 26: 3D reconstruction of the **Room** scene by the structure from motion technique described in this paper: back projection on the first camera and projection on the ground plane

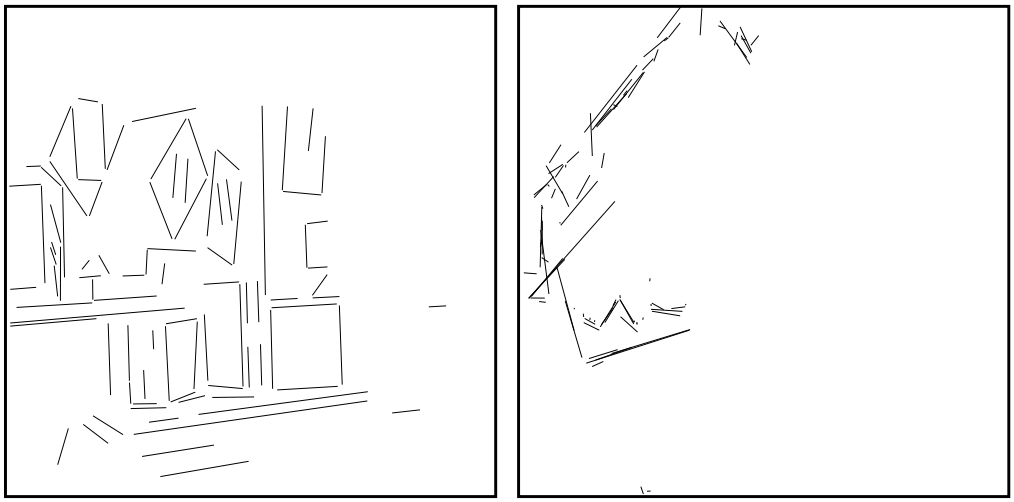

Fig. 27: 3D reconstruction of the **Room** scene by a classical trinocular stereo: back projection on the first camera and projection on the ground plane

ding to the two false matches is easily identified, from the top views, to be the isolated line segments near the bottom edge of the picture. The reconstruction with our algorithm is noisier than that with the trinocular stereo, but in the latter, three images are used giving more constraints in 3D reconstruction.

The third set of real data is an image pair of a scene named **Modig** because it contains a painting by the Italian painter Modigliani (see Fig. 28). There are 121 line segments matched by the trinocular stereo (see Fig. 29). As usual, there exist a few false matches. One can also notice several multiple matches: several segments on the painting are fragmented in one view, and the fragments are matched to a single segment in the other view.

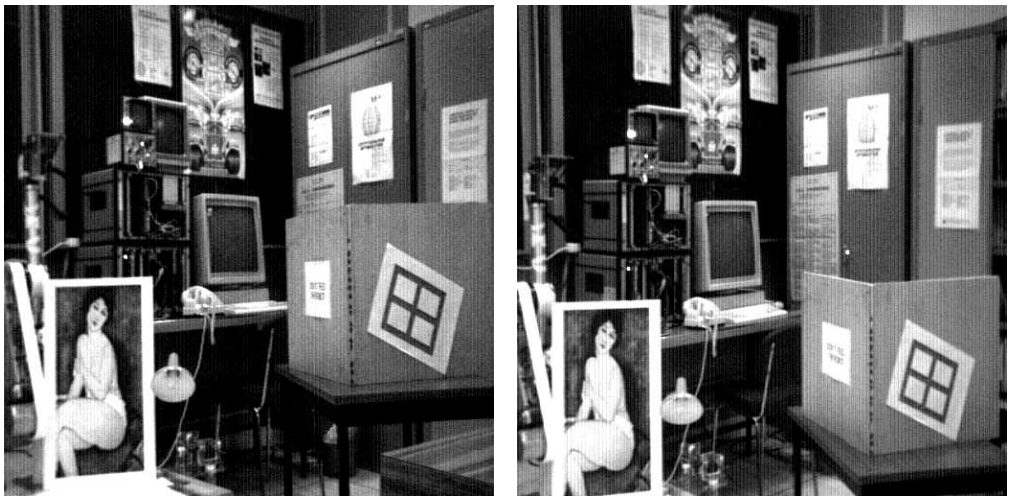

Fig. 28: Image pair of a **Modig** scene

Again, we searched for the initial motion estimate by sampling. The ten best samples all converge to the good solution. The final motion estimation is:

$$\mathbf{r} = [8.481e{-}2, -8.545e{-}2, 2.064e{-}2]^T \ , \ \mathbf{t} = [4.847e{-}1, 8.665e{-}1, 1.199e{-}1]^T \ ,$$

while the estimation through stereo calibration is:

$$\mathbf{r} = [6.250e{-}2, -7.196e{-}2, 2.109e{-}2]^T \ , \ \mathbf{t} = [4.757e{-}1, 8.661e{-}1, 1.536e{-}1]^T \ .$$

The difference in the rotation angle is $1.406°$; the angle between the rotation axes is $4.642°$; the angle between the translation vectors is $2.002°$.

The 3D reconstruction produced by our algorithm is shown in Fig. 26, while that produced by the trinocular stereo is shown in Fig. 27. The two results are comparable. Again, due to use of the *union* strategy, our 3D reconstruction is more complete.

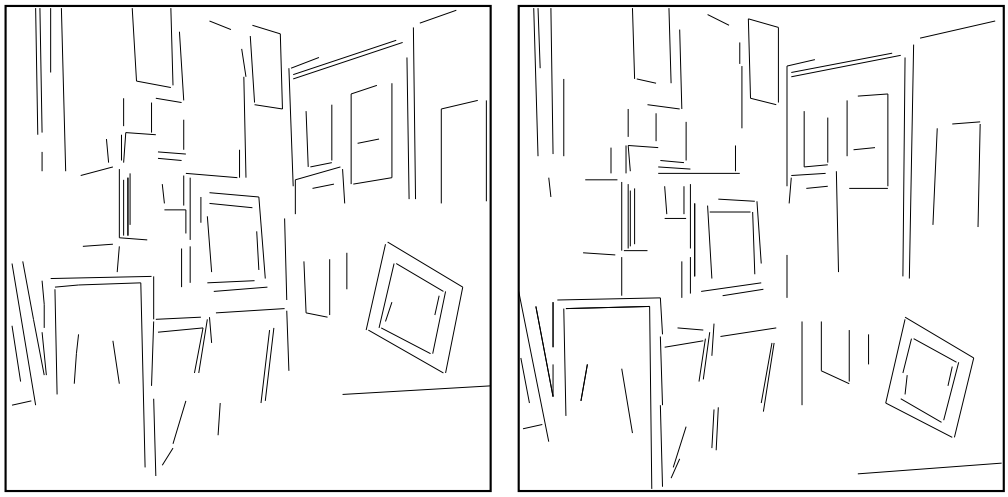

Fig. 29: Matched line segments of the **Modig** image pair

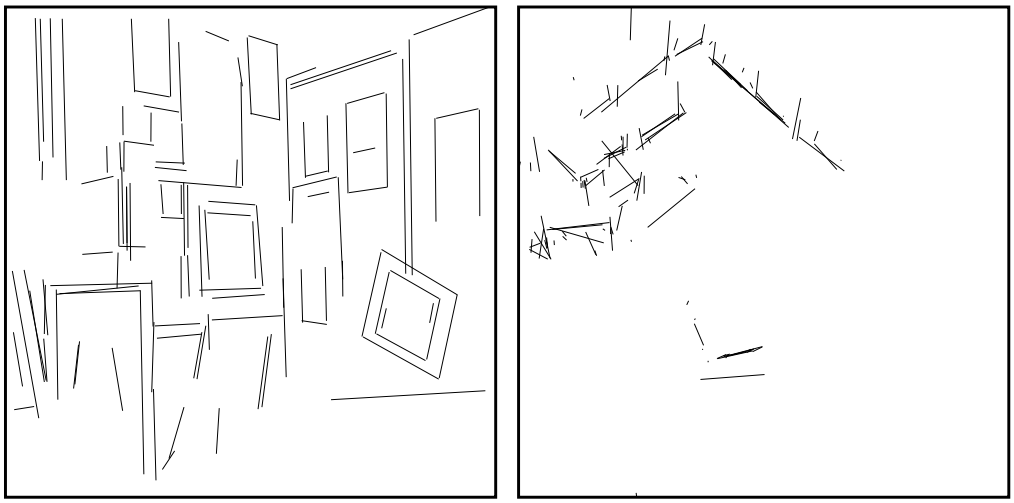

Fig. 30: 3D reconstruction of the **Modig** scene by the structure from motion technique described in this paper: back projection on the first camera and projection on the ground plane

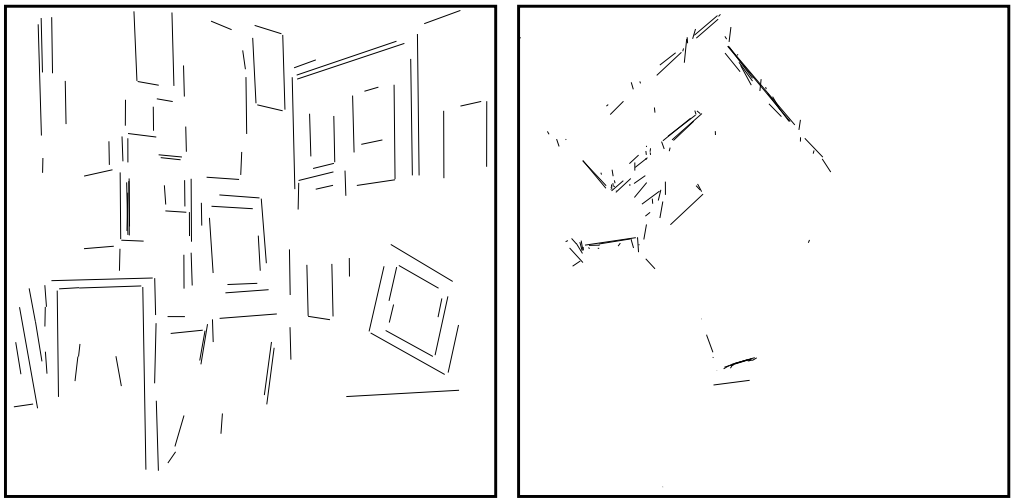

Fig. 31: 3D reconstruction of the **Modig** scene by a classical trinocular stereo: back projection on the first camera and projection on the ground plane

6 Conclusions

We have shown for the first time that the 3D motion and structure can be determined from two perspective images using only *line segments*. Classical methods use their geometric abstraction, namely straight lines, but then *three images* are necessary. The algorithm we proposed in this paper is based on the assumption that two matched line segments contain the projection of a *common part* of the corresponding line segment in space. Indeed, this is what we use to match line segments between different views. This assumption has been implemented by using the epipolar geometry. Because a closed-form solution is not available, we have proposed to sample the motion space, which is five-dimensional. Both synthetic and real data have been used to test the proposed algorithm, and excellent results have been obtained with real data containing about one hundred line segments. The results are comparable with those obtained with calibrated stereo.

From our experiences, we observe that in order to obtain a usable result we need a relatively large set of correspondences of line segments (say, 50). To alleviate this requirement, we can use points in combination, and this is what we are doing.

Recently, many researchers are working with uncalibrated images using points and straight lines [5, 10, 14, 20]. The proposed algorithm in this paper can be easily extended to estimate the epipolar geometry between two uncalibrated images using line segments. The only problem is that we need to sample a higher parameter space (seven dimensions now) to find an initial estimate.

Acknowledgment

The sample data of a Gaussian sphere was kindly supplied by Luc Robert. Charlie Rothwell checked the English and made several comments.

References

[1] N. Ayache and F. Lustman. Fast and reliable passive trinocular stereovision. In *Proc. First Int'l Conf. Comput. Vision*, pages 422–427, London, UK, June 1987. IEEE.

[2] D. H. Ballard and C. M. Brown. *Computer Vision.* Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] J.K. Cheng and T.S. Huang. Image registration by matching relational structures. *Pattern Recog.*, 17(1):149–159, 1984.

[4] R. Deriche and O. Faugeras. Tracking line segments. In O. Faugeras, editor, *Proc. First European Conf. Comput. Vision*, pages 259–268, Antibes, France, April 1990. Springer, Berlin, Heidelberg.

[5] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig. In Giulio Sandini, editor, *Proceedings of the 2nd European Conference on Computer Vision*, pages 563–578. Springer-Verlag, Lecture Notes in Computer Science 588, May 1992.

[6] O. D. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, Cambridge, MA, 1993.

[7] O.D. Faugeras and G. Toscani. The calibration problem for stereo. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 15–20, Miami, FL, June 1986. IEEE.

[8] B. Giai-Checa, R. Deriche, T. Vieville, and O. Faugeras. Suivi de segments dans une séquence d'images monoculaire. Technical Report 2113, INRIA Sophia-Antipolis, France, December 1993.

[9] G. Giraudon. Chaînage efficace contour. Rapport de Recherche 605, INRIA, Sophia-Antipolis, France, February 1987.

[10] R. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, pages 761–764, Urbana, 1992.

[11] T.S. Huang and A.N. Netravali. Motion and structure from feature correspondences: A review. *Proc. IEEE*, 82(2):252–268, February 1994.

[12] Y. Liu and T.S. Huang. A linear algorithm for determining motion and structure from line correspondences. *Comput. Vision, Graphics Image Process.*, 44(1):35–57, 1988.

[13] H.C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, 1981.

[14] Q.-T. Luong. *Matrice fondamentale et calibration visuelle sur l'environnement: Vers une plus grande autonomie des systèmes robotiques.* Dissertation, University of Paris XI, Orsay, Paris, France, December 1992.

[15] J.W. Roach and J.K. Aggarwal. Determining the movement of objects from a sequence of images. *IEEE Trans. PAMI*, 2(6):554–562, 1980.

[16] M. Spetsakis and J. Aloimonons. A unified theory of structure from motion. Technical Report CAR-TR-482, Computer Vision Laboratory, University of Maryland, December 1989.

[17] M. E. Spetsakis and J. Aloimonos. Structure from Motion Using Line Correspondences. *Int'l J. Comput. Vision*, 4:171–183, 1990.

[18] R.Y. Tsai and T.S. Huang. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surface. *IEEE Trans. PAMI*, 6(1):13–26, January 1984.

[19] S. Ullman. *The Interpretation of Visual Motion.* MIT Press, Cambridge, MA, 1979.

[20] T. Viéville, Q.T. Luong, and O.D. Faugeras. Motion of points and lines in the uncalibrated case. *Int'l J. Comput. Vision*, 1994. To appear.

[21] Z. Zhang and O. Faugeras. *3D Dynamic Scene Analysis: A Stereo Based Approach.* Springer, Berlin, Heidelberg, 1992.

Unité de recherche INRIA Lorraine, Technopôle de Nancy-Brabois, Campus scientifique,
615 rue du Jardin Botanique, BP 101, 54600 VILLERS LÈS NANCY
Unité de recherche INRIA Rennes, Irisa, Campus universitaire de Beaulieu, 35042 RENNES Cedex
Unité de recherche INRIA Rhône-Alpes, 46 avenue Félix Viallet, 38031 GRENOBLE Cedex 1
Unité de recherche INRIA Rocquencourt, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex
Unité de recherche INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, 06902 SOPHIA-ANTIPOLIS Cedex

Éditeur
INRIA, Domaine de Voluceau, Rocquencourt, BP 105, 78153 LE CHESNAY Cedex (France)
ISSN 0249-6399