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

Zhengyou Zhang

***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 of 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 for the motion and structure determination process. In this paper we show that it is possible to recover motion from two views 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, i.e., they overlap. Indeed, this is what we use to match line segments between different views. This assumption constrains the possible motion between two views to an open set in motion parameter space. A heuristic, consisting of maximizing the overlap, leads to a unique solution. Both synthetic and real data have been used to test the proposed algorithm, and excellent results have been obtained with real data containing a relatively large set of line segments.**

***Index Terms*—Motion, structure from motion, line segments, epipolar geometry, perspective images, overlap, dynamic scene analysis.**

I. 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 [22] assumed an orthographic camera projection model and showed that three views are necessary to recover the motion and structure from point correspondences; Roach and Aggarwal [18] 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 [1], [12] for a complete review, and to [16] for a theoretical study.

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 [14], [21] or for line correspondences [20], [13]. Algorithms using both points and lines are also available [19]. However, another important type of geometric primitives, namely that of *line segments*, has been since long ignored in motion and structure from motion[1], although 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 [5], [9] or other techniques [4]. This paper is organized as follows. Section II describes the problem we want to solve and shows why we can recover 3D motion and structure from corresponding line segments between two images. Section III 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 IV addresses the issue of 3D reconstruction of line segments provided the motion is estimated. Section V provides the experimental results with real data. Section VI terminates the paper with several discussions.

II. 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.

Manuscript received Oct. 25, 1994; revised May 31, 1995. Recommended for acceptance by S. Peleg.

The author is with INRIA Sophia-Antipolis, 2004 route des Lucioles, BP 93, F-06902 Sophia-Antipolis Cedex, France; e-mail: zzhang@sophia.inria.fr.

To order reprints of this article, e-mail: transactions@computer.org, and reference IEEECS Log Number P95141.

[1] 3D line segments, reconstructed by a stereo system, have been used in motion analysis by Zhang and Faugeras [25], but the problem there is different from the one addressed here.

0162-8828/95$04.00 © 1995 IEEE

A. Notation

We use bold low case letters **a**, **b**, **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 **A**, **B**, **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 **a**, and $\mathbf{A}^T$ is the transpose of **A**. The cross product of two vectors **a** and **b** is denoted by $\mathbf{a} \times \mathbf{b}$. The dot product of **a** and **b** is denoted by $\mathbf{a} \cdot \mathbf{b}$ or $\mathbf{a}^T\mathbf{b}$.

The coordinates of a point **a** in image plane are $[u, v]^T$; for the reason that will become clear sooner, we use $\tilde{\mathbf{a}} = [u, v, 1]^T$, i.e., adding 1 as the last element to **a**. Similarly, for a point $M = [x, y, z]^T$ in 3D space, we have $\tilde{M} = [x, y, z, 1]^T$.

Additional notation will be introduced in the following subsections.

B. Geometry of the Motion Problem of Line Segments

We consider a calibrated camera, which is modeled as a standard pinhole. The relation between each point M in space and its corresponding point **m** in image plane is linear projective, and is described by a perspective transformation, i.e.,

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

where s is an arbitrary scalar 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.

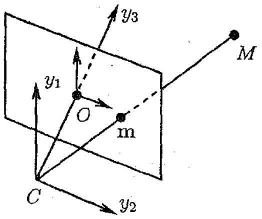


Fig. 1. The pinhole camera model.

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.

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 $(\mathbf{R}, \mathbf{t})$: given a 3D point **x** expressed in the coordinate frame associated to the first camera, it is equal to $\mathbf{Rx} + \mathbf{t}$ in that associated to the second camera.

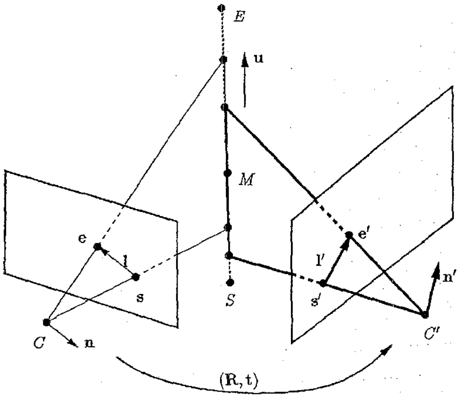


Fig. 2. The geometry of the motion of line segments.

An image line segment **l** is represented by its starting point **s** and its endpoint **e**. We assume that line segments are *orientated*, which can be obtained from the intensity contrast information. For example, we can define an orientation signing convention such that the intensity changes from a low value to a high one when we cross the line segment from left to right. As we observe later, the orientation information is dispensable in the motion and structure problem. However, in general this information is available, and has usually already been used in the matching process.

We are given two line segments, **l** in camera 1 and **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 (**s** and **s**$'$) and the endpoints (**e** and **e**$'$) are in correspondence. Indeed, these points are not reliable, mainly for three reasons:

1) 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.
2) 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.
3) 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.

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 [10].

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.

C. 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, we use $\tilde{\mathbf{a}} = [u, v, 1]^T$ to denote the same point in $Cy_1y_2y_3$ coordinate frame, and its projective coordinates will be $\lambda\tilde{\mathbf{a}}$ for any nonzero scalar λ. Without loss of generality, $\tilde{\mathbf{a}}$ is also used to denote the projective coordinates. For any point $\mathbf{m} = [u, v]^T$ on the infinite supporting line $\mathbf{l}$, we have the following relation:

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

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 [7]: the line defined by two points $\mathbf{a}$ and $\mathbf{b}$ is represented by $\mathbf{l} = \tilde{\mathbf{a}} \times \tilde{\mathbf{b}}$; the intersection point of two lines $\mathbf{l}_1$ and $\mathbf{l}_2$ is represented by $\tilde{\mathbf{m}} = \mathbf{l}_1 \times \mathbf{l}_2$. Dividing the first two elements of $\tilde{\mathbf{m}}$ 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 $\mathbf{n} = \tilde{\mathbf{s}} \times \tilde{\mathbf{e}}$.

D. Unlike Supporting Lines, Line Segments Can Constrain the Motion

It has been well known that motion cannot be determined from two views of straight lines [13], [7]. 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. By requiring a pair of matched line segments to overlap in space, we add a constraint on the family of feasible motions. The set of all constraints for all correspondences of line segments define an open set in motion parameter space. The larger the number of correspondences is, the smaller the extent of the open set is, and the more the motion is constrained. If we have only a few correspondences of line segments, the motion might not be well constrained and the corresponding reconstruction of the scene geometry will vary widely.

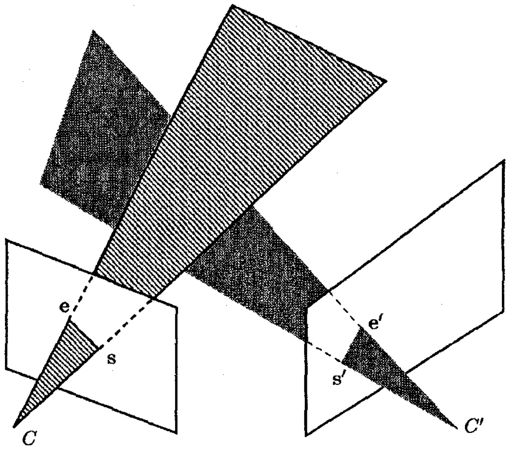


Fig. 3. Motion and structure from line segments.

III. 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.

A. Epipolar Constraint

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

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

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

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

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 **m** must lie. The line $\mathbf{l}'_\mathbf{m}$ is known as the *epipolar line* of **m**. The above constraint is known as the *epipolar constraint.*

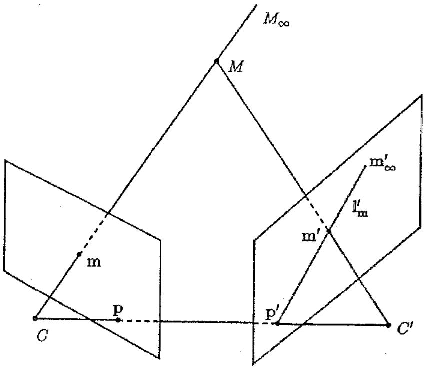


Fig. 4. Epipolar geometry.

The epipolar line can be defined by two points. The first point can be obtained by projecting M' with $\lambda = 0$, which gives $\tilde{\mathbf{p}}' = \frac{1}{t_3}\mathbf{t}$, where t_3 is the third element of the translation vector **t**, or projectively, $\tilde{\mathbf{p}}' = \mathbf{t}$. 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 $\tilde{\mathbf{m}}'_\infty = \frac{1}{\mathbf{r}_3^T\tilde{\mathbf{m}}}\mathbf{R}\tilde{\mathbf{m}}$, where $\mathbf{r}_3$ is the third row of the rotation matrix **R**, or projectively $\tilde{\mathbf{m}}'_\infty = \mathbf{R}\tilde{\mathbf{m}}$. As described in Section II.C, the epipolar line $\mathbf{l}'_\mathbf{m}$ is projectively represented by

$$\mathbf{l}'_\mathbf{m} = \tilde{\mathbf{p}}' \times \tilde{\mathbf{m}}'_\infty = \mathbf{t} \times \mathbf{R}\tilde{\mathbf{m}}. \qquad (2)$$

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 **x**. Letting $\mathbf{E} = [\mathbf{t}]_\times\mathbf{R}$, (2) can be rewritten as

$$\mathbf{l}'_\mathbf{m} = \mathbf{E}\tilde{\mathbf{m}}. \qquad (3)$$

The matrix **E** is the well-known *essential matrix* [14].

It is easy to see that all epipolar lines in the second image pass through the single point $\mathbf{p}'$. 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 $\mathbf{p}'$ which is the projection of C. The point $\mathbf{p}'$ is thus called the *epipole* in the second image.

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

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

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

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

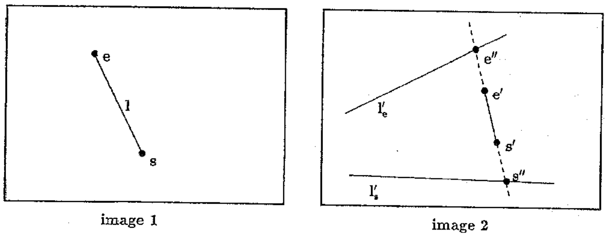


Fig. 5. Overlap of two line segments in correspondence

Provided that the epipolar geometry (i.e., matrix **E**, or the motion (**R**, **t**)) between two images is correct, then **s** and $\mathbf{s}''$ correspond to a single point in space; so do **e** and $\mathbf{e}''$. Thus, the statement that two line segments **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, one of the following two conditions must be satisfied:

1) Either $\mathbf{s}''$ or $\mathbf{e}''$ or both are between $\mathbf{s}'$ and $\mathbf{e}'$.
2) $\mathbf{s}'$ and $\mathbf{e}'$ are both between $\mathbf{s}''$ and $\mathbf{e}''$.

This implies that only when (here $\|$ stands for the **or** logic)

$$\begin{aligned} &(\mathbf{s}''-\mathbf{s}')\cdot(\mathbf{e}'-\mathbf{s}'') > 0 \,\|\, (\mathbf{e}''-\mathbf{s}')\cdot(\mathbf{e}'-\mathbf{e}'') > 0 \\ &\|\, (\mathbf{s}'-\mathbf{s}'')\cdot(\mathbf{e}''-\mathbf{s}') > 0 \,\|\, (\mathbf{e}'-\mathbf{s}'')\cdot(\mathbf{e}''-\mathbf{e}') > 0, \end{aligned} \qquad (4)$$

the two line segments $\mathbf{s}'\mathbf{e}'$ and $\mathbf{s}''\mathbf{e}''$ overlap.

The above constraint does not use the fact that the line segments are oriented. The configuration in Fig. 6 satisfies the above constraint, but does not satisfy the orientation congruence. If the rotation between two images is not very big, then the orientation of the projected line segment in image cannot change abruptly. 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:

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

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 two constraints are satisfied for each correspondence. However, we can only obtain a feasible *region* in the motion space, and we rather need a unique solution. In the next, we solve the problem by maximizing the overlap of line segments.

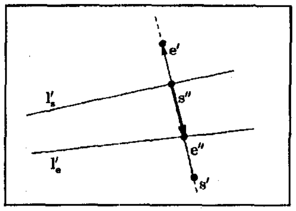


Fig. 6. Two incongruent line segments in orientation.

C. 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 two constraints 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.

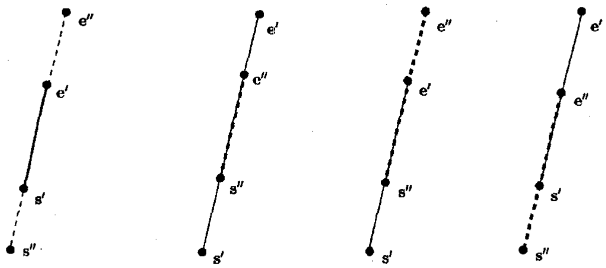


Fig. 7. Four configurations of two line segments with overlap.

The overlap length, denoted by $\mathcal{L}'$, is defined as the length of the common segment, which is given by

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

If two segments do not overlap (i.e., the inequalities in Section III.B are not satisfied), we define the overlap length as

$$\mathcal{L}' = -\min(\|\mathbf{e}' - \mathbf{s}''\|, \|\mathbf{e}'' - \mathbf{s}'\|), \qquad (6)$$

which corresponds to the gap between the two segments. The reader is referred to [24] for more details.

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, $\{(\mathbf{l}_i, \mathbf{l}_i') | i = 1, \ldots, n\}$, estimate the camera motion parameters $(\mathbf{R}, \mathbf{t})$ by minimizing the following objective function

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

D. Implementation Details

The minimization of the objective function (6) is conducted using a *downhill simplex method* [17].

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 equal to the rotation angle. The vector $\mathbf{r}$ is related to the matrix $\mathbf{R}$ by the Rodrigues formula [25]:

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

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

Because the magnitude of $\mathbf{t}$ is inherently unrecoverable, the translation $\mathbf{t}$ may be assumed to be of unit length, and hence is represented by a point on the unit sphere. The spherical coordinates (ϕ, θ) is used to represent $\mathbf{t}$.

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 endpoints or midpoints, but the results are useless. For the solution that works best, we choose to sample the parameter space to obtain a global minimum. The space of rotation can be thought of as a solid ball of radius π. Assume that the motion between two successive views is small, we sample the range $\left[-\frac{\pi}{4}, \frac{\pi}{4}\right]$ with step equal to $\frac{\pi}{8}$ in each direction. The maximum rotation angle is $\frac{\sqrt{3}}{4}\pi$ (i.e., 78°). This range is sufficient for most applications of motion analysis. It is rare that the rotation angle goes beyond 60° between successive views. We have thus $5^3 = 125$ samples of rotation.[2]

The samples of translation are obtained through a uniform partition of a Gauss sphere based on the icosahedron [3]. 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

[2] This sampling is not uniform. A better way may be the following. The solid ball of radius π is first sampled by a set of spheres of radius from 0 to π with step equal to, say, $\frac{\pi}{8}$. Each sphere can then be quasi-uniformly sampled as we will do for translation. This technique is inspired from the comments of an anonymous reviewer.

them, i.e., the samples from a hemisphere.

PROPOSITION 1. *We consider two given 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

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

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 hemisphere 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 of the camera coordinate frame because $\det(-\mathbf{R}) = -1$. They are thus excluded on physical grounds.

To summarize, we have $125 \times 40 = 5{,}000$ 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.

It is well-known that the relative position of two cameras is determined up to a π radians twist about the line joining the two optical centers [16]. The reader may wonder why we do not consider this ambiguity in the above sampling technique. The reason is that we have assumed a small rotation (less than $\pi/2$) between two cameras. In that case the twisted solution never appears in our sampling range because its corresponding rotation is always bigger than $\pi/2$, and it is not a physical solution. If we do not restrict the rotation angle between the two cameras, then we need to consider the twisted pair ambiguity to find the physical solution. As suggested by one of the reviewers, the property of the twisted pair can be used to cut the number of rotations by half, thus halving the search space.

IV. 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

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

The solution is:

$$\mathbf{x} = \left[\mathbf{n}\ \ \mathbf{R}^T\mathbf{n}'\ \ \mathbf{n} \times \left(\mathbf{R}^T\mathbf{n}'\right)\right]^{-T} \begin{bmatrix} 0 \\ -\mathbf{t}^T\mathbf{n}' \\ 0 \end{bmatrix}.$$

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 Section II.B, 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. In passing, our trinocular stereo algorithm [2] uses the *intersection* strategy, that is, only the part of line segment which is perceived by all of the three cameras is reconstructed in space.

V. EXPERIMENTAL RESULTS

The proposed algorithm has been tested with both synthetic and real data. The reader is referred to [24] for the results with synthetic data.

For comparison reason, we have also tried to apply a point-based method [7] to the endpoints or midpoints of line segments, but the results are useless. For example, for the **Modig** scene described below (Figs. 8-12), the motion estimated when applying the point-based method to the endpoints is:

$$\mathbf{r} = [-2.244e-2, -2.284e-2, 1.224e-4]^T,$$
$$\mathbf{t} = [3.577e-1, 8.368e-1, 4.146e-1]^T,$$

which is completely different from that obtained through stereo calibration:

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

The difference in the rotation angle is 5.164°, but the angle between the rotation axes is as high as 89.25°, and the angle between the translation vectors is 18.55°. The 3D reconstruction provided by the point-based method is meaningless, and is thus not shown here. This is why we resort to the sampling technique described in Section III.D for searching an initial estimate of motion.

At the time of writing, we have tested our algorithm with success on more than ten real image pairs, except for one which contains many line segments *aligned with the epipolar lines* (the latter case is well known to cause problems for binocular stereo). This is because our algorithm computes the overlap from the intersections of line segments with their corresponding epipolar lines, and the intersections will be unstable when they are almost aligned. One solution to this would be to compute the angles between line segments and their corresponding epipolar lines, and we could just discard those line segments that form a small angle with the epipolar lines.

In the following, we describe four sets of real data which were extracted from a trinocular stereo system [2]. We have chosen the stereo data because the stereo system has been calibrated which serves as a ground truth [8].

The first 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. 8). There are 121 line segments matched by the trinocular stereo (see Fig. 9), among which 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.

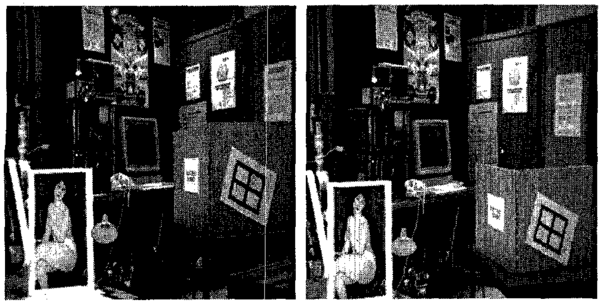


Fig. 8. Image pair of a **Modig** scene.

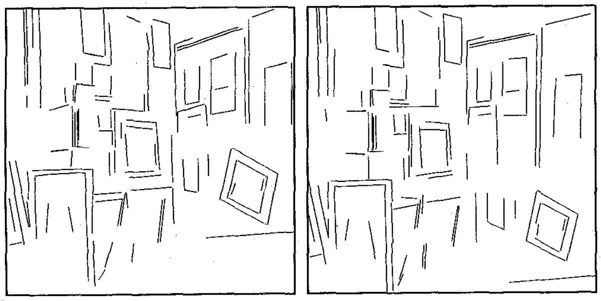


Fig. 9. Matched line segments of the **Modig** image pair.

We searched for the initial motion estimate by sampling as described in Section III.D. The 10 best samples all converge to the good solution. The best solution is the one which minimizes (6). The final motion estimation is:

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

which should be compared with the estimation through stereo calibration already given at the beginning of this section. 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°.

To better understand how the proposed algorithm works, we have extracted the intermediate results for one selected hypothesis of motion. Recall that our algorithm tries to maximize the overlap of two sets of line segments. Fig. 10 shows how the overlap of the two sets of line segments of the **Modig** scene evolves during the optimization process. The four pictures correspond to the results obtained with the initial motion estimate and those obtained after five, 10, and 24 iterations. The first set of line segments are projected onto the second image using the motion estimate as described in Section III.B, and are shown in Fig. 10 as solid lines. The second set of line segments are shown as dashed lines. The matched line segments are shown to be collinear because of the way of the projection performed. As can be observed, the overlap by the initial motion estimate is very bad. Significant improvement is achieved after five iterations. Very good result is already obtained after ten iterations. Later on, the improvement is small, as can be asserted from the comparison of the two pictures of the lower row in Fig. 10.

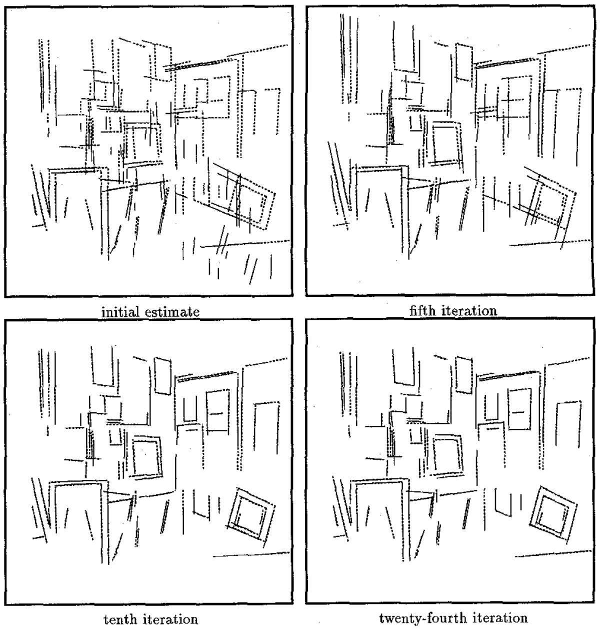


Fig. 10. Evolution of the overlap of the line segments of the **Modig** scene during the optimization process: They correspond to the results of the initial, fifth, tenth and 24th iteration.

The 3D reconstruction produced by the algorithm described in this paper is shown in Fig. 11, where the picture on the left is the perspective view from the first camera and the one on the right is a top view. This result should be compared with that reconstructed by our trinocular stereo which uses *three* images (note that only two images are used for the other algorithm) and whose geometry has been previously calibrated (see Fig. 12). The two results are comparable. Because of use of the *union* strategy described in Section IV, the 3D reconstruction shown in Fig. 11 appears more complete than that shown in Fig. 12.

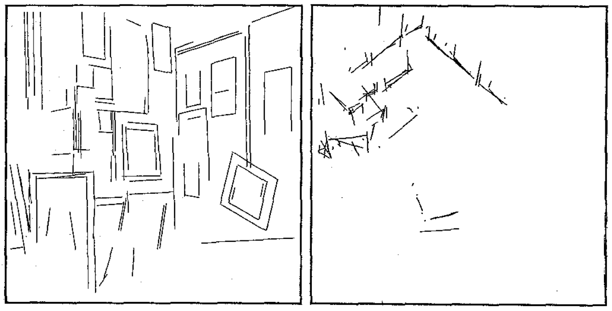


Fig. 11. 3D reconstruction of the **Modig** scene by the structure from motion technique described in this paper: back projection on the first camera and projection on the ground plane.

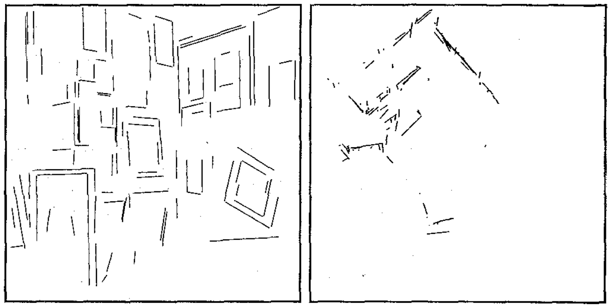


Fig. 12. 3D reconstruction of the **Modig** scene by a classical trinocular stereo: back projection on the first camera and projection on the ground plane.

The second set of real data is an image pair of a **RobotLab** scene (see Fig. 13). Forty-five line segments have been matched by our trinocular stereo system,[3] as shown in Fig. 14. One can notice several false matches.

Through searching for the initial motion estimation by sampling as described in Secttion III.D, six among the 10 best samples converge to the good solution. The motion estimation given by the algorithm described in this paper is:

$$\mathbf{r} = [1.859e-1,\ 1.218e-1,\ 3.707e-2]^T,$$
$$\mathbf{t} = [-5.939e-1,\ 6.341e-1,\ -4.951e-1]^T,$$

while the estimation through stereo calibration is:

$$\mathbf{r} = [1.927e-1,\ 1.195e-1,\ 3.407e-2]^T,$$
$$\mathbf{t} = [-5.927e-1,\ 7.325e-1,\ -3.349e-1]^T.$$

The difference in the rotation angle is 0.228°; the angle between the rotation axes is 1.682°; the angle between the translation vectors is 10.788°. The translation is not very well estimated. The overlap of the two sets of line segments given by the final motion estimate is shown in Fig. 15.

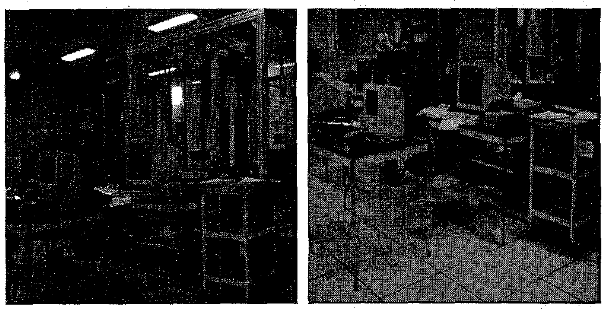


Fig. 13. Image pair of a **RobotLab** scene.

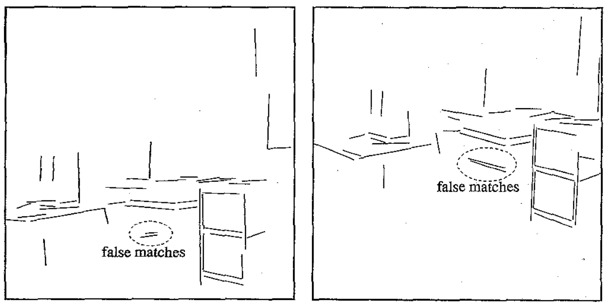


Fig. 14. Matched line segments of the **RobotLab** image pair.

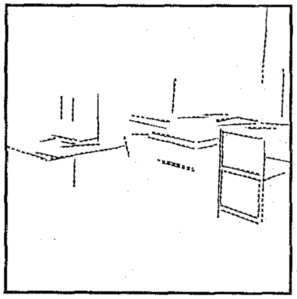


Fig. 15. Overlap of the two sets of line segments given by the final motion estimate.

The projection on the ground plane of the 3D reconstruction based on the technique described in Section IV is displayed on the left in Fig. 16. This result should be compared with that reconstructed by our trinocular stereo which uses *three* images and whose geometry has been previously calibrated (see the picture on the right in Fig. 16). Essentially, almost the same result can be observed. The two very long, almost vertical line segments in the left picture of Fig. 16 correspond to two false matches on the floor. As we use the *union* strategy in 3D reconstruction, the reconstruction of the false matches becomes outstanding.

[3] The stereo can in fact match more line segments, but we have imposed a tighter epipolar constraint to limit the number of matches for this example.

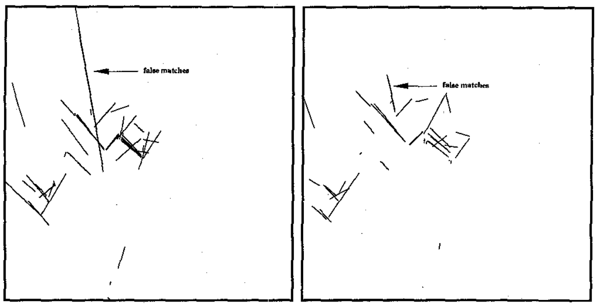


Fig. 16. Comparison of the 3D reconstruction of the **RobotLab** scene by the structure from motion technique described in this paper (left) and that by a classical trinocular stereo (right): projection on the ground plane.

The third set of real data is an image pair of a scene named **Room** (see Fig. 17). Ninety line segments have been matched by our trinocular stereo system, as shown in Fig. 18. One can easily notice two false matches located at the lower right corner near the border of the table.

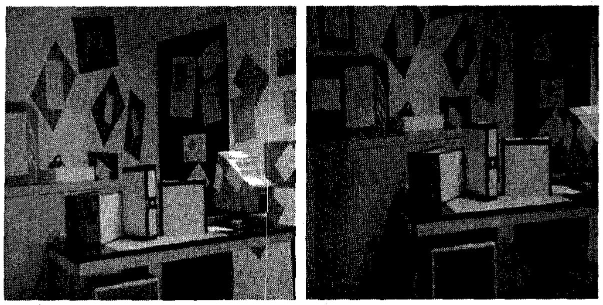


Fig. 17. Image pair of a **Room** scene.

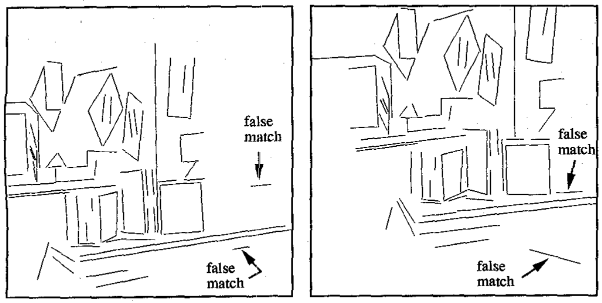


Fig. 18. Matched line segments of the **Room** image pair.

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:

$$\mathbf{r} = [1.124e-1, 1.807e-1, 1.850e-2]^T,$$
$$\mathbf{t} = [-7.939e-1, 5.904e-1, -1.451e-1]^T,$$

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 1.445°; the angle between the rotation axes is 1.839°; the angle between the translation vectors is 2.366°. The overlap of the two sets of line segments given by the final motion estimate is shown in Fig. 19.

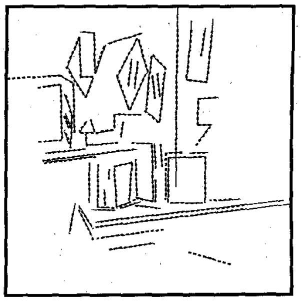


Fig. 19. Overlap of the two sets of line segments given by the final motion estimate.

The projection on the ground plane of the 3D line segments reconstructed by our algorithm are shown on the left in Fig. 20, while those reconstructed by the trinocular stereo are shown on the right in Fig. 20. The reconstruction corresponding to the two false matches is easily identified to be the isolated line segments near the bottom edge of the picture.

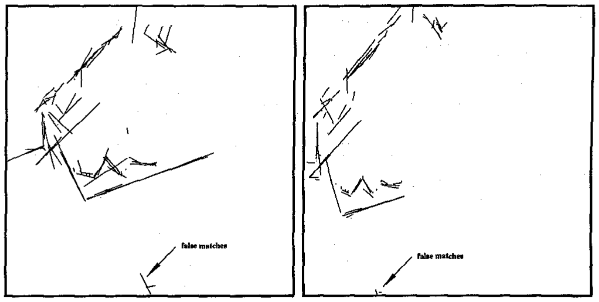


Fig. 20. Comparison of the 3D reconstruction of the **Room** scene by the structure from motion technique described in this paper (left) and that by a classical trinocular stereo (right): projection on the ground plane.

The fourth set of real data is an image pair of a scene named **Table** because it contains a turntable (see Fig. 21). There are 128 line segments matched by the trinocular stereo (see Fig. 22). As usual, there exist a few false matches. We also notice that the border of the turntable is differently segmented because of curvature.

Again, we searched for the initial motion estimate by sampling. For each sample, we evaluated the cost function (6), and ten best samples were retained for further optimization. Five among these 10 samples converge to the good solution. The final motion estimation is:

$$\mathbf{r} = [-8.924e-3, 2.194e-2, 2.224e-2]^T,$$
$$\mathbf{t} = [-5.517e-1, -7.523e-1, -3.601e-1]^T,$$

while the estimation through stereo calibration is:

$$\mathbf{r} = [-8.520e-3, 2.660e-2, 2.140e-2]^T,$$
$$\mathbf{t} = [-5.784e-1, -7.355e-1, -3.527e-1]^T.$$

The difference in the rotation angle is 0.155°; the angle between the rotation axes is 6.643°; the angle between the translation vectors is 1.856°. The overlap of the two sets of line segments given by the final motion estimate is shown in Fig. 23.

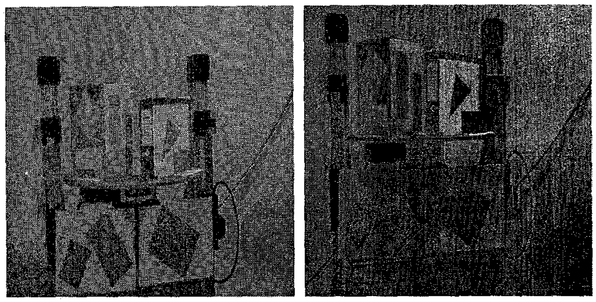


Fig. 21. Image pair of a **Table** scene.

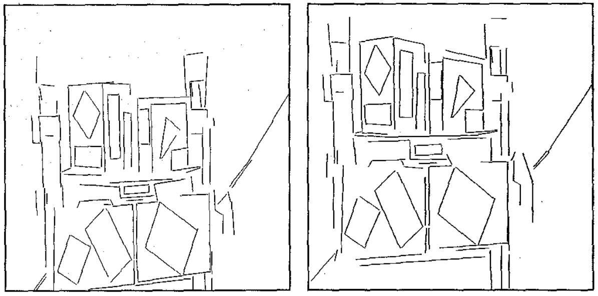


Fig. 22. Matched line segments of the **Table** image pair.

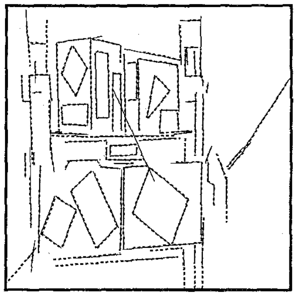


Fig. 23. Overlap of the two sets of line segments given by the final motion estimate.

The projection on the ground plane of the 3D reconstruction produced by our algorithm is shown on the left in Fig. 24, while that produced by the trinocular stereo is shown on the right in Fig. 24. The two results are comparable, but the 3D reconstruction corresponding to false matches with our method is easier to be remarked, e.g., the two very long, almost vertical line segments in the middle. This is because we use the union strategy in 3D reconstruction, while the trinocular stereo reconstructs only the part common to all three images which usually produces much shorter 3D line segments for false matches.

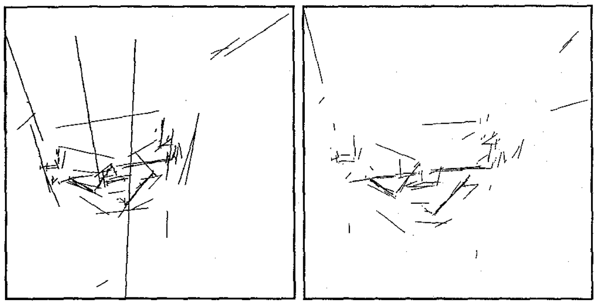


Fig. 24. Comparison of the 3D reconstruction of the **Table** scene by the structure from motion technique described in this paper (left) and that by a classical trinocular stereo (right): projection on the ground plane.

VI. Summary and Discussions

We have shown for the first time that both 3D motion and structure can be computed from two perspective images using only *line segments*. Classical methods use their geometric abstraction, namely straight lines, but then *three images* are necessary. We have tried to apply a now well-known *point-based* method [7] to the endpoints or midpoints of line segments. However, because of the instability of these points due to the reasons described in Section II.B, we have not obtained any meaningful results with all of the real line segments that were tried. 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 through the use of the epipolar geometry, which is of course unknown. Because a closed-form solution is not available, we have proposed a solution which samples 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 stereo calibration.

As described in Section II.D, by requiring a pair of matched line segments to overlap in space, we add a constraint on the family of feasible motions. It is this constraint that allows the computation of motion from correspondences of two sets of line segments. If we have only a small set of matches, the feasible motion may not be well constrained, and the result will be poor. From our experience, 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 and line segments in combination, and the investigation of this issue is currently under way.

The proposed algorithm tries to find the motion by maximizing the overlap of line segments. This is a heuristic and a biased estimate may be obtained if the variation of the line segments is not random. When a large set of line segments are used, this problem is not severe because the motion is well constrained, as we have seen from the results with real data.

Another problem we encountered with the proposed algorithm, as already mentioned in Section V, arises when line constrained, as we have seen from the results with real data.

Another problem we encountered with the proposed algorithm, as already mentioned in Section V, arises when line segments are aligned with the epipolar lines. This is because the overlap is computed from the intersection of line segments with their epipolar lines, and the intersections are instable when they are almost parallel. This case is also well-known to cause problems for binocular stereo. One solution to this would be just to discard the line segments that form a small angle with the epipolar lines.

Recently, many researchers have worked with uncalibrated images using points or straight lines [6], [11], [15], [23]. The algorithm proposed in this paper can easily be 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.

Acknowledgments

The sample data of a Gaussian sphere was kindly supplied by Luc Robert. Charlie Rothwell checked the English and made several comments. The author enjoyed the discussion with Steve Maybank on the twisted pair ambiguity. The anonymous reviewers are gratefully acknowledged for their constructive comments which have helped me to improve this paper.

References

[1] J.K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—a review," *Proc. IEEE*, vol. 76, no. 8, pp. 917–935, Aug. 1988.

[2] N. Ayache and F. Lustman, "Fast and reliable passive trinocular stereovision," *Proc. First Int'l Conf. Computer Vision*, pp. 422–427, London, June 1987.

[3] D.H. Ballard and C.M. Brown, *Computer Vision*. Englewood Cliffs, N.J.: Prentice Hall, 1982.

[4] J.K. Cheng and T.S. Huang, "Image registration by matching relational structures," *Pattern Recognition*, vol. 17, no. 1, pp. 149–159, 1984.

[5] R. Deriche and O. Faugeras, "Tracking line segments," *Proc. First European Conf. Computer Vision*, O. Faugeras, ed., pp. 259–268, Antibes, France, Apr. 1990.

[6] O.D. Faugeras, "What can be seen in three dimensions with an uncalibrated stereo rig," *Proc. Second European Conf. Computer Vision*, Giulio Sandini, ed., pp. 563–578, *Lecture Notes in Computer Science 588*, Springer-Verlag, May 1992.

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

[8] O.D. Faugeras and G. Toscani, "The calibration problem for stereo," *Proc. IEEE Conf. Computer Vision and Pattern Recog.*, pp. 15–20, Miami, June 1986.

[9] 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, Dec. 1993.

[10] G. Giraudon, "Chaînage efficace contour," Rapport de Recherche 605, INRIA, Sophia-Antipolis, France, Feb. 1987.

[11] R. Hartley, R. Gupta, and T. Chang, "Stereo from uncalibrated cameras," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 761–764, Urbana, Ill., 1992.

[12] T.S. Huang and A.N. Netravali, "Motion and structure from feature correspondences: A review," *Proc. IEEE*, vol. 82, no. 2, pp. 252–268, Feb. 1994.

[13] Y. Liu and T.S. Huang, "A linear algorithm for determining motion and structure from line correspondences," *Comput. Vision, Graphics Image Processing*, vol. 44, no. 1, pp. 35–57, 1988.

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

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

[16] S.J. Maybank, *Theory of Reconstruction from Image Motion*. Springer-Verlag, 1992.

[17] J.A. Nelder and R. Mead, "A simplex method for function minimization," *Computer J.*, no. 7, pp. 308–313, 1965.

[18] J.W. Roach and J.K. Aggarwal, "Determining the movement of objects from a sequence of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 2, no. 6, pp. 554–562, 1980.

[19] M. Spetsakis and J. Aloimonons, "A unified theory of structure from motion," Technical Report CAR-TR-482, Computer Vision Laboratory, Univ. of Maryland, Dec. 1989.

[20] M.E. Spetsakis and J. Aloimonos, "Structure from motion using line correspondences," *Int'l J. Computer Vision*, vol. 4, pp. 171–183, 1990.

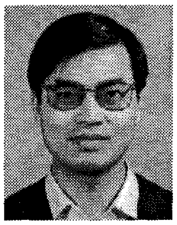
[21] R.Y. Tsai and T.S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surface," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol 6, no. 1, pp. 13–26, Jan. 1984.

[22] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, Mass.: MIT Press, 1979.

[23] T. Viéville, Q.T. Luong, and O.D. Faugeras, "Motion of points and lines in the uncalibrated case," *Int'l J. Computer Vision*, to appear, vol. 17, no. 1.

[24] Z. Zhang, "Estimating motion and structure from correspondences of line segments between two perspective images," Research Report 2340, INRIA Sophia, 1994.

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



Zhengyou Zhang received the BS degree in electronic engineering from the University of Zhejiang, China, in 1985, the DEA diploma in computer science from the University of Nancy, France, in 1987, the PhD degree in computer science from the University of Paris XI, Orsay, France, in 1990, and the diploma *Habilitation à diriger des recherches* from the same university in 1994.

From 1987 to 1990, he was a research assistant in the Computer Vision and Robotics Group of the French National Institute for Research in Computer Science and Control (INRIA), France. He is currently a senior research scientist at INRIA. Dr. Zhang's current research interests include computer vision, mobile robotics, dynamic scene analysis, and multisensor fusion. He is the coauthor (with O. Faugeras) of the book *3D Dynamic Scene Analysis: A Stereo Based Approach* (Springer-Verlag, 1992) and (with S. Ma) of the forthcoming book *Computer Vision* (in Chinese) (Chinese Academy of Sciences, 1996).