Estimation of Displacements from Two 3-D Frames Obtained from Stereo

Zhengyou Zhang and Olivier D. Faugeras, *Senior Member, IEEE*

Abstract— **We present a method for estimating 3-D displacements from two stereo frames. It is based on the hypothesize-and-verify paradigm used to match 3-D line segments between the two frames. In order to reduce the complexity of the method, we make the assumption that objects are rigid. We formulate a set of complete rigidity constraints for 3-D line segments and integrate the uncertainty of measurements in this formulation. The hypothesize-and-verify stages of the method use an extended Kalman filter to produce estimates of the displacements and of their uncertainty. In the experimental sections, the algorithm is shown to work on indoor and natural scenes. Furthermore, it is easily extended, as is also shown, to the case where several mobile objects are present. The method is quite robust, fast, and has been thoroughly tested on hundreds of real stereo frames.**

Index Terms— **Extended Kalman filtering, hypothesize-and-verify, motion from stereo, multiple object motions, rigidity constraints, robot vision, 3-D matching, uncertainty.**

I. Introduction

MOTION ANALYSIS is a very important research field in robot vision and is essential for the interpretation of 3-D dynamic scenes. Its applications include mobile robot navigation, scene segmentation, construction of a world model, dynamic surveillance, and object tracking. Most previous research efforts have been on the motion analysis of a sequence of monocular images [1]–[3]. With the development of stereovision systems and range finders [4]–[9], the computation of motion from 3-D data becomes a more attractive technique. Our research focuses on motion analysis from 3-D frames that are obtained at different instants by a stereo system when the mobile robot navigates in an unknown environment possibly containing some moving rigid objects. The features reconstructed by our stereo are 3-D-oriented line segments. The orientation is obtained from the image intensity contrast.

The problem of determining the 3-D structure and motion of objects from image data has been extensively studied by computer scientists for more than a decade. Table I summarizes the current approaches to motion analysis. They are first distinguished from each other according to whether they use either the changes of feature positions or those of gray levels between adjacent frames. The first approach is usually called *feature tracking*, and the second is known as *optical flow*. In each case, one can consider the number of frames used: short sequence analysis (two or three frames) or long sequence analysis (more than four). One can further classify the current approaches according to the dimension of the available data. If we have only one image at each instant, we have only 2-D data, and the problem is called monocular sequence analysis. If we have two or three images at each instant, we can reconstruct 3-D data using stereo, and the problem is called binocular or trinocular sequence analysis. We can thus roughly divide existing methods into eight categories.

TABLE I
Summary of Approaches to Motion Analysis

changes	sequence	dimension
features	**short** (2 or 3)	**2-D**
gray levels	**long** ($\geq$ 4)	**3-D**

We do not intend to review all previous work here. The reader is referred to [10]–[15] for the work related to the optical flow approach, to [16]–[22] for the work related to the motion estimation from 2-D feature correspondences over two or three frames, and to [23]–[28] for the work related to (2-D or 3-D) feature tracking approaches using a large number of frames. Those references are by no means exhaustive. The work reported in this paper falls into the category of motion analysis using 3-D data from stereo over two frames.

The problem is usually tackled in two steps. The first step is to establish feature correspondences between two frames. The correspondence (matching) problem is recognized as a very difficult one. The rigidity assumption provides a powerful constraint to simplify the analysis and is used in almost all matching algorithms. Pollard *et al.* [29] describe a matching strategy combining the local feature focus method proposed in [30] and [31] and the constrained tree-search technique using rigidity constraints proposed in [32]. Chen and Huang [33] propose an approach to match two sets of 3-D points, which also exploits the rigidity constraints. Due to the completeness of the defined constraints for 3-D points, that approach avoids extensive comparisons of interrelationships and model tests like in the general tree-search approaches. The same authors presented, in [34], an algorithm for matching 3-D line segments. Using the point-matching algorithm described in [33], the directions of line segments can be matched. Several potential matchings may exist. For each potential match, a rotation can be computed. In the second stage, a Hough-like procedure is used to prune those potential matchings and to compute a 3-D translation that brings the (rotated) line segments into correspondences. Kim and Aggarwal [35] use a relaxation method to register two 3-D frames from stereo. The

Manuscript received October 15, 1990; revised November 26, 1991. This work was supported by Esprit project P940. Recommended for acceptance by Associate Editor W. E. L. Grimson.

The authors are with INRIA, Valbonne, France.

IEEE Log Number 9202757.

constraints used are the distance and angle between a line and its neighbor. Once the feature correspondences are established, the second step is to estimate the 3-D motion between the two frames. Many methods exist in the literature, including analytic or numeric, linear or nonlinear, batch, or recursive ones [36]–[40]. Due to the uncertainty in the measurements, these methods minimize some criteria (usually least squares).

Under some restricted conditions, motion can be computed without registration between two frames. In [41], for example, 3-D points are used, and the same set of points is assumed to be observed to undergo the same motion between successive views. These assumptions, of course, are not realistic. Features may be visible in one view and not in another view due to, for example, occlusion, appearance, or disappearance. Multiple moving objects may also exist in the scene.

It is worth noticing that motion analysis shares many common points with another important research field in computer vision: object recognition and localization. Grimson and Lozano-Perez proposed in [32] and [42] a tree pruning approach using rigidity constraints to recognize 2-D objects from 2-D data or 3-D objects from 3-D data. For example, they used distance and angular constraints to match measured 3-D points with model faces. To ensure the global consistency, a model test is required. Their approach suffers from exponential combinatorics in the case of spurious data and occluded objects. The scheme of the algorithm presented in this paper is similar to those reported in [38] and [43] in the sense that all methods fall into the "hypothesize-and-verify" paradigm. Faugeras and Hébert proposed, in [38], a method to match points and planar patches from range data with a list of planar patches in a stored model. Ayache and Faugeras described, in [43], a method to match 2-D line segments from polygonal approximation of an image to 2-D objects in a model base that is also represented by 2-D line segments. A similar approach is proposed by Lowe in [44] and [45] to recognize 3-D objects from 2-D images. Lowe made use of perceptual organization to form structures in the image that are likely viewpoint invariant. Evidential reasoning was used to establish the initial feature correspondences. He iteratively updated the viewpoint estimation by adding more feature correspondences.

This paper deals with the analysis of motion or, rather, of displacement from two stereo views. In [46] and [47], we reported our preliminary work on this problem. This paper includes new developments, gives a more complete description of our approach, and provides new results. Since rigidity may be the most important constraint in two-view motion analysis, we first provide a thorough description of it. We show that the rigidity constraints we formulate are complete for 3-D line segments. The rigidity constraints are reformulated to take into account explicitly the uncertainty in measurements. We then develop an approach based on the *hypothesize-and-verify* paradigm for registering two stereo frames and computing the 3-D displacement between them. The rigidity constraints formulated before are heavily used to generate hypotheses of primitive correspondences between two frames. We propose a new representation for 3-D line segments in order to characterize the uncertainty caused by stereo. Using this representation, 3-D displacement can be incrementally estimated via an extended Kalman filter, and matching can be done efficiently. Finally, we provide two experimental examples and the application of this algorithm to multiple object motions.

II. Outline of the Matching Algorithm

The matching problem has been recognized as a very difficult problem. Given two sets of primitives observed in two views, the task of matching is to establish a correspondence between them. By a correspondence, we mean that the two paired primitives are the different observations (instances) of a single primitive undergoing motion. The matching problem has an exponential complexity in general. For example, if we structure the matching as a search in the interpretation tree [32], one should examine the consistency of $(n+1)^m$ interpretations, where n and m are the numbers of primitives in the two views. Recognizing this, we need some constraints to heuristically guide the search without exploring most of the nodes while making sure a good interpretation is found at the end.

The rigidity assumption about the environment and objects is used in most matching algorithms. Our approach uses that rigidity constraint to guide a hypothesize-and-verify method. The hypothesize-and-verify paradigm is one of the most popular paradigms in dealing with the matching problem [29]–[32], [38], [43].

Our idea is simple. We use the rigidity constraints to generate some hypothetical primitive correspondences between two successive frames. We compute an initial estimate of the displacement for each hypothesis. We then evaluate the validity of these hypothetical displacements. Due to the rigidity constraints (see below), the number of hypothetical displacements is usually very small, and computational efficiency is achieved. We exploit the rigidity constraint locally in the hypothesis generation phase and globally in the hypothesis verification phase, as will be shown later. Fig. 1 illustrates diagrammatically the principle of our hypothesize-and-verify method.

III. Rigidity Constraints

A. 3-D Rigid Displacement

It is well known that any 3-D rigid displacement can be *uniquely* decomposed into a rotation around the origin of the coordinate system followed by a translation. Let P and P' be the position vectors of the same 3-D point before and after displacement, where the following relation holds:

$$P' = \boldsymbol{R}P + \boldsymbol{t}, \tag{1}$$

where $\boldsymbol{R}$ is called the *rotation matrix*, and $\boldsymbol{t}$ is called the *translation vector* of the rigid displacement. $(\boldsymbol{R}, \boldsymbol{t})$ must satisfy the following requirements.

Definition 1: $(\boldsymbol{R}, \boldsymbol{t})$ represents a *rigid displacement* if and only if the following requirements are satisfied:

- $\boldsymbol{R}$ is a 3×3 orthogonal matrix, and its determinant equals $+1$.
- $\boldsymbol{t}$ is a real 3-D vector.

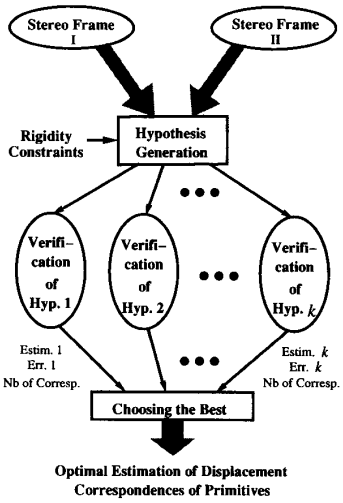


Fig. 1. Diagram of the displacement analysis algorithm based on the hypothesize-and-verify paradigm.

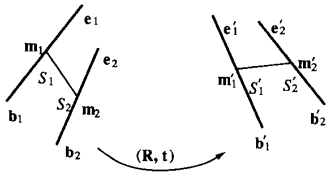


Fig. 2 Line segments undergoing a rigid displacement.

Proposition 1 gives some interesting properties of the rotation matrix $\boldsymbol{R}$ (the proof is omitted).

Proposition 1: The rotation matrix $\boldsymbol{R}$ has the following properties:

1. $\boldsymbol{R}\boldsymbol{R}^T = \boldsymbol{R}^T\boldsymbol{R} = \boldsymbol{I}_3$, and $\boldsymbol{R}^{-1} = \boldsymbol{R}^T$
2. $\|\boldsymbol{R}\boldsymbol{u}\| = \|\boldsymbol{u}\|$
3. $\boldsymbol{R}\boldsymbol{u}\cdot\boldsymbol{R}\boldsymbol{v} = \boldsymbol{u}\cdot\boldsymbol{v}$
4. $\boldsymbol{R}\boldsymbol{u}\wedge\boldsymbol{R}\boldsymbol{v} = \boldsymbol{R}(\boldsymbol{u}\wedge\boldsymbol{v})$

where $\boldsymbol{u}$ and $\boldsymbol{v}$ are two arbitrary 3-D vectors, ^{T} denotes the transpose of a matrix, $^{-1}$ denotes the inverse of a matrix, $\|\ \|$ denotes the norm of a vector, $\cdot$ denotes the inner product of two vectors, and $\wedge$ denotes the cross product of two vectors.

Consider now two line segments undergoing a rigid displacement (see Fig. 2). The two segments are denoted by S_1 and S_2 before displacement and by S_1' and S_2' after displacement. We use the following notations: The unit direction vector of S is denoted by $\boldsymbol{u}$, and the endpoints are denoted by $\boldsymbol{b}$ and $\boldsymbol{e}$ (*Remark*: Segments are oriented). If the rigid displacement is represented by $(\boldsymbol{R}, \boldsymbol{t})$, we have the following equations:

$$\begin{aligned} \boldsymbol{u}_i' &= \boldsymbol{R}\boldsymbol{u}_i, \quad \boldsymbol{m}_i' = \boldsymbol{R}\boldsymbol{m}_i + \boldsymbol{t}, \\ \boldsymbol{b}_i' &= \boldsymbol{R}\boldsymbol{b}_i + \boldsymbol{t}, \quad \boldsymbol{e}_i' = \boldsymbol{R}\boldsymbol{e}_i + \boldsymbol{t}, \quad \text{for } i = 1, 2. \end{aligned} \tag{2}$$

B. Rigidity Constraints

Under rigid displacement, the geometry of a rigid object remains constant. In other words, the geometry of the object does not change during displacement. We want to derive specific constraints reflecting this invariance. The following requirements should be satisfied:

- The constraints should be independent of the coordinate systems. The relation between the coordinate systems is just what we want to compute.
- The constraints should be as complete as possible to guarantee the global consistency of the final interpretation.
- The constraints should be as simple as possible to allow efficient computation.

The following proposition gives necessary conditions.

Proposition 2: If two segments S_1 and S_2 in the first frame are matched to two segments S_1' and S_2' in the second frame, under the assumption of rigid displacement, the following constraints, which are known as *rigidity constraints*, must be satisfied:

1. *Length constraints*

$$l_1' = l_1 \quad \text{and} \quad l_2' = l_2 \tag{3}$$

2. *Distance constraint*

$$\|\boldsymbol{v}_{12}\| = \|\boldsymbol{v}_{12}'\| \tag{4}$$

3. *Angular constraints*

$$\boldsymbol{u}_1\cdot\boldsymbol{u}_2 = \boldsymbol{u}_1'\cdot\boldsymbol{u}_2' \tag{5}$$

$$\boldsymbol{u}_1\cdot\hat{\boldsymbol{v}}_{12} = \boldsymbol{u}_1'\cdot\hat{\boldsymbol{v}}_{12}' \tag{6}$$

$$\boldsymbol{u}_2\cdot\hat{\boldsymbol{v}}_{12} = \boldsymbol{u}_2'\cdot\hat{\boldsymbol{v}}_{12}' \tag{7}$$

4. *Triple product constraint*

$$\langle\boldsymbol{u}_1, \boldsymbol{u}_2, \hat{\boldsymbol{v}}_{12}\rangle = \langle\boldsymbol{u}_1', \boldsymbol{u}_2', \hat{\boldsymbol{v}}_{12}'\rangle. \tag{8}$$

In the above, $\langle\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3\rangle$ denotes the triple product (i.e., $\boldsymbol{u}_1\cdot(\boldsymbol{u}_2\wedge\boldsymbol{u}_3)$), $\boldsymbol{v}_{12}$ is the vector joining the midpoints (i.e., $\boldsymbol{v}_{12} = \boldsymbol{m}_2 - \boldsymbol{m}_1$), and $\boldsymbol{v}_{12}$ is a unit length vector parallel to $\boldsymbol{v}_{12}$ (i.e., $\hat{\boldsymbol{v}}_{12} = \boldsymbol{v}_{12}/\|\boldsymbol{v}_{12}\|$).

The above proposition can be easily verified using (2) and the properties of rigid displacement described in Proposition 1 (see [48] for the proof).

In the rigidity constraints, three angles are involved (θ, θ_1, and θ_2 in Fig. 3). In this figure, the segment $//S_2$ is parallel to segment S_2 and shares a common endpoint with S_1. Instead of angles, we use the cosine of angles. Equation (5) implies the conservation of the angle between S_1 and S_2 (i.e., the angle θ between S_1 and $//S_2$). Equation (6) implies the conservation of the angle θ_1 between S_1 and the segment $\boldsymbol{v}_{12}$ joining the midpoints. Equation (7) implies the conservation of the angle θ_2 between S_2 and $\boldsymbol{v}_{12}$.

The reverse of Proposition 3.2 is proved in [48].

Proposition 3: The rigidity constraints given in Proposition 2 are complete for 3-D line segments in the sense that they are necessary and sufficient to guarantee the congruency of two sets of line segments.

The following proposition follows.

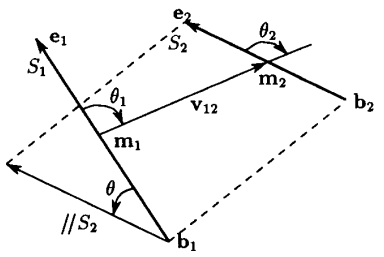


Fig. 3. Definition of the angles used in rigidity constraints.

Proposition 4: The displacement computed from two pairings of noncollinear segments satisfying the rigidity constraints in Proposition 2 is rigid and unique.

Since two pairings of noncollinear segments satisfying the rigidity constraints implies the congruency of two tetrahedra (or quadrilaterals) from Proposition 3, a unique rigid displacement can be computed (see [48] for the proof).

IV. Representation of 3-D Line Segments

Before proceeding further, we briefly describe in this section how to represent 3-D line segments. The 3-D line segments we have (from stereo or other sensors) are inherently uncertain and usually do not have the same error distribution in different directions. It has been recognized in the computer vision and robotics community [49], [50] that uncertainty should be explicitly represented and manipulated.

A. Motivation

A line is usually represented by its endpoints M_1 and M_2, which require six parameters and their covariance matrices Λ_1 and Λ_2. Λ_1 and Λ_2 are estimated from stereo triangulation from point correspondences [51]. Equivalently, a line segment can be represented by its direction vector $\boldsymbol{v}$, its midpoint M, and their covariance matrices. However, we cannot use directly these parameters in most cases. The endpoints or the midpoint of a segment are not reliable. The reason for this is that the way the uncertainty of the endpoints of a 3-D segment is computed takes only into account the uncertainty in pixel location due to edge detection and the uncertainty in the calibration of the stereo rig [52]. It does not, however, take into account the uncertainty due to the variations in segmentation of the polygonal approximation process. There are two main sources for those variations. The first one is purely algorithmic; because of noise in the image and because we sometimes approximate curved contours with straight line segments, the polygonal approximation may vary from frame to frame, inducing a variation in the segments endpoints yet unaccounted for. The second is physical; because of partial occlusion in the scene, a segment can be considerably shortened or lengthened, and this also has to be taken into account in the modeling of uncertainty.

Thus, instead of the line segment, the infinite line supporting the segment is usually used, as in [35]. In an earlier version of our algorithm, for displacement analysis from two stereo views [40], [46], [47], a line segment was treated in a mixed way. The infinite supporting line was used in estimating the displacement, and the line segment was used for matching. There has been a lot of representations proposed for a line in the literature [50], [53]. The main problem is that *the uncertainty on the line parameterization does not reflect that of the segment the line supports* (see [48] for more details). A segment with large uncertainty may result in small uncertainty in the line parametrization. In the next subsection, we describe a representation for 3-D line segments, taking into account the variations in segmentation.

B. Our Representation

Due to the deficiencies of the previous representations for a line or a line segment, we use a five-parameter representation for a line segment: two for the orientation, and three for the position of a point on the segment. This is a tradeoff between line and segment. If we add another parameter for the length, the line segment is fully specified. Special care is given to the representation of uncertainty.

1) Representing the Orientation by Its Euler Angles ϕ and θ: Consider the spherical coordinates. Let $\boldsymbol{u} = [u_x,\ u_y,\ u_z]^T$ be a unit vector of orientation. We have

$$\begin{cases} u_x = \cos\phi \sin\theta \\ u_y = \sin\phi \sin\theta \\ u_z = \cos\phi \end{cases} \tag{9}$$

with $0 \le \phi < 2\pi, \quad 0 \le \theta \le \pi$.

From $\boldsymbol{u}$, we can compute $\phi,\ \theta$

$$\phi = \begin{cases} \arccos \frac{u_x}{\sqrt{1-u_z^2}} & \text{if } u_y \ge 0 \\ 2\pi - \arccos \frac{u_x}{\sqrt{1-u_z^2}} & \text{otherwise} \end{cases}$$
$$\theta = \arccos u_z. \tag{10}$$

If we denote $[\phi,\ \theta]^T$ by $\boldsymbol{\Psi}$, then the mapping between $\boldsymbol{\Psi}$ and $\boldsymbol{u}$ is 1-to-1 except when $\theta = 0$. When $\theta = 0$, ϕ is not defined. Fortunately, the component corresponding to ϕ in the covariance matrix computed below will be very large in this case, as shown in [48]. This means that the measurement ϕ has no information. Another problem with this representation is the discontinuity of ϕ when a segment varies nearly parallel to the plane $y = 0$. In that case, the angle ϕ may jump from the interval $[0,\ \pi/2)$ to the interval $(3\pi/2,\ 2\pi)$ or *vice versa*. This discontinuity must be dealt with in matching and integration.

In the following, we assume that the direction vector $\boldsymbol{v} = [x,\ y,\ z]^T$ and its covariance matrix $\Lambda_{\boldsymbol{v}}$ of a given segment are known. We want to compute $\boldsymbol{\Psi}$ and its covariance matrix Λ_Ψ from $\boldsymbol{v}$ and $\Lambda_{\boldsymbol{v}}$. From (11), ϕ and θ are simply given by

$$\phi = \begin{cases} \arccos \frac{x}{\sqrt{x^2+y^2}} & \text{if } y \ge 0 \\ 2\pi - \arccos \frac{x}{\sqrt{x^2+y^2}} & \text{otherwise} \end{cases}$$
$$\theta = \arccos \frac{z}{\sqrt{x^2+y^2+z^2}}. \tag{11}$$

Since the relation between $\boldsymbol{\Psi}$ and $\boldsymbol{v}$ is not linear, we use the first-order approximation to compute the covariance matrix

$\Lambda_{\boldsymbol{\Psi}}$ from $\Lambda_{\boldsymbol{v}}$, that is

$$\boldsymbol{\Lambda}_{\boldsymbol{\Psi}} = \frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{v}} \boldsymbol{\Lambda}_{\boldsymbol{v}} \frac{\partial \boldsymbol{\Psi}^T}{\partial \boldsymbol{v}} \tag{12}$$

where the Jacobian matrix

$$\frac{\partial \boldsymbol{\Psi}}{\partial \boldsymbol{v}} = \begin{bmatrix} \frac{\partial \phi}{\partial x} & \frac{\partial \phi}{\partial y} & \frac{\partial \phi}{\partial z} \\ \frac{\partial \theta}{\partial x} & \frac{\partial \theta}{\partial y} & \frac{\partial \theta}{\partial z} \end{bmatrix}.$$

2) Modeling the Midpoint of a 3-D Line Segment: We choose the midpoint as the point on the segment, but a special treatment of the covariance is introduced to characterize the uncertainty in the location of a segment.

The midpoint M and its covariance matrix Λ_M can be computed from the endpoints of the segment by

$$M = (M_1 + M_2)/2 \text{ and } \Lambda_M = (\Lambda_1 + \Lambda_2)/4.$$

As described earlier, the measurement of the endpoints is not reliable. The position of a segment may vary along its direction in different views. Therefore, we model the midpoint $\boldsymbol{m}$ as follows:

$$\boldsymbol{m} = (M_1 + M_2)/2 + n\boldsymbol{u} \tag{13}$$

where $\boldsymbol{u}$ is the unit direction vector of the segment, and n is a random scalar independent of M_1 and M_2. Equation (13) says, in fact, that the midpoint of a segment is only partially credible; it may vary randomly along the line supporting it in successive views. As explained before, the main reason of this modelization is that the uncertainty measures of M_1 and M_2 given by the stereo system modelize only the uncertainties due to stereo triangulation but not those due to, for example, different segmentations. Remark that this modelization is in accordance with the definition of a line. If a point $\boldsymbol{p}_0$ on a line and its orientation $\boldsymbol{u}$ are given, the line L may be defined as a set of points in 3-D space parametrized by a real variable t:

$$L = \{\boldsymbol{p} \mid \boldsymbol{p} = \boldsymbol{p}_0 + \boldsymbol{u}t, \ -\infty < t < \infty\}. \tag{14}$$

The random variable n in (13) is modeled as Gaussian zero-mean with deviation σ_n, which is a positive scalar. If a segment is reliable, σ_n may be chosen to be a small number; if not, it may be chosen to be a big number. In our implementation, σ_n is related to the length l of the segment, i.e., $\sigma_n = \kappa l$, where κ is some constant independent of the segment. That is to say, if a segment is longer, the deviation σ_n is bigger. That is reasonable since a long segment is very likely to be broken into smaller segments in other views due to, for example, occlusion. In our experiments, $\kappa = 0.2$.

In order to compute the covariance of $\boldsymbol{m}$, we should first compute $\boldsymbol{u}$ and $\Lambda_{\boldsymbol{u}}$. The unit direction vector $\boldsymbol{u}$ and its covariance $\Lambda_{\boldsymbol{u}}$ can be computed from the nonnormalized direction vector $\boldsymbol{v}$ and its covariance matrix $\Lambda_{\boldsymbol{v}}$. Indeed, we have

$$\boldsymbol{u} = \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}, \qquad \Lambda_{\boldsymbol{u}} = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{v}} \Lambda_{\boldsymbol{v}} \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{v}}^T \tag{15}$$

where $\partial \boldsymbol{u}/\partial \boldsymbol{v}$ is a 3×3 matrix ($\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{v}} = \frac{\boldsymbol{I}_3}{\|\boldsymbol{v}\|} - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|^3}$). Note that the covariance matrix $\Lambda_{\boldsymbol{u}}$ is singular (the determinant is zero). This is, of course, because the three components of $\boldsymbol{u}$ are not independent.

At this point, the covariance of $\boldsymbol{m}$ can be computed. Equation (16) can be rewritten as

$$\boldsymbol{m} = M + n\boldsymbol{u}. \tag{16}$$

Since n and $\boldsymbol{u}$ are independent of each other, we have

$$E[n\boldsymbol{u}] = E[n]E[\boldsymbol{u}] = 0 \tag{17}$$

$$\Lambda_{n\boldsymbol{u}} = E[(n\boldsymbol{u})(n\boldsymbol{u})^T] = E[n^2]E[\boldsymbol{u}\boldsymbol{u}^T] = \sigma_n^2(\Lambda_{\boldsymbol{u}} + \bar{\boldsymbol{u}}\bar{\boldsymbol{u}}^T) \tag{18}$$

where $\bar{\boldsymbol{u}}$ denotes $\mathrm{E}[\boldsymbol{u}]$, and

$$\mathrm{E}[\boldsymbol{m}] = \mathrm{E}[M], \tag{19}$$

$$\Lambda_{\boldsymbol{m}} = \mathrm{E}[(\boldsymbol{m} - \mathrm{E}[\boldsymbol{m}])(\boldsymbol{m} - \mathrm{E}[\boldsymbol{m}])^T] = \Lambda_M + \Lambda_{n\boldsymbol{u}}. \tag{20}$$

If we add another parameter l to denote the length of the segment, we can then represent a line segment exactly. This ends our modelization of a line segment (see [54] and [48] for more details).

V. Error Measurements in the Constraints

In Section III, we have proposed a number of rigidity constraints for 3-D line segments and shown that they were complete (necessary and sufficient conditions for the existence of a rigid displacement). We have also shown that we can compute a unique rigid displacement from two pairings of segments satisfying the rigid constraints. However, all derivations were based on the noise-free assumption. The data we have are always corrupted with noise due to uncertainties in polygonal approximation, stereo calibration, and reconstruction. The equalities in Proposition 2 are no longer true. In this section, we reformalize the rigidity constraints by explicitly taking into account the uncertainty of measurements. The idea is to *dynamically* compute a threshold for each constraint from the uncertainty in 3-D data as modeled in the last section. The reader is referred to [32], [29], [34], and [55] for other formalisms of rigidity constraints in the literature. Unlike our approach, all those approaches use some predefined error ranges in the measurements. However, the errors of measurements given by a stereo system have different distributions in different positions. One cannot handle such phenomena with some prefixed values.

Examining Proposition 2, we find that the rigidity constraints take three forms:

Norm constraint

$$\|\boldsymbol{v}\| = \|\boldsymbol{v}'\| \tag{21}$$

Dot–product constraint

$$\boldsymbol{u} \cdot \boldsymbol{v} = \boldsymbol{u}' \cdot \boldsymbol{v}' \tag{22}$$

Triple–product constraint

$$< \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3 > = < \boldsymbol{u}_1', \boldsymbol{u}_2', \boldsymbol{u}_3' > . \tag{23}$$

A. Norm Constraint

The norm constraint says that the difference between the norms of two vectors should be zero. For convenience, we use the squared norm of a vector instead of its norm. Taking into account the uncertainty of measurements, we can formalize the constraint as follows:

$$| \|\boldsymbol{v}\|^2 - \|\boldsymbol{v}'\|^2 | < \varepsilon_n \tag{24}$$

where ε_n is the threshold to be determined on the norm constraint.

Replacing $\|\boldsymbol{v}\|^2 - \|\boldsymbol{v}'\|^2$ by d_n, we then have

$$d_n = \boldsymbol{v} \cdot \boldsymbol{v} - \boldsymbol{v}' \cdot \boldsymbol{v}' = \boldsymbol{v}^T \boldsymbol{v} - \boldsymbol{v}'^T \boldsymbol{v}'. \tag{25}$$

Given the covariance matrix $\Lambda_{\boldsymbol{v}}$ of $\boldsymbol{v}$ and the covariance matrix $\Lambda_{\boldsymbol{v}'}$ of $\boldsymbol{v}'$, we now compute the variance Λ_{d_n} of d_n. Under the first-order approximation, we have

$$\Lambda_{d_n} = J_{\boldsymbol{v}}^{d_n} \Lambda_{\boldsymbol{v}} J_{\boldsymbol{v}}^{d_n\,T} + J_{\boldsymbol{v}'}^{d_n} \Lambda_{\boldsymbol{v}'} J_{\boldsymbol{v}'}^{d_n\,T}, \tag{26}$$

where $J_{\boldsymbol{v}}^{d_n}$ is the Jacobian matrix of d_n with respect to $\boldsymbol{v}$ and $J_{\boldsymbol{v}'}^{d_n}$, which is the one with respect to $\boldsymbol{v}'$. Here, we assume that $\boldsymbol{v}$ and $\boldsymbol{v}'$ are two independent Gaussian random variables.

The Jacobian $J_{\boldsymbol{v}}^{d_n}$ is given by $J_{\boldsymbol{v}}^{d_n} = \frac{\partial d_n}{\partial \boldsymbol{v}} = \frac{\partial}{\partial \boldsymbol{v}}(\boldsymbol{v}^T\boldsymbol{v} - \boldsymbol{v}'^T\boldsymbol{v}') = 2\boldsymbol{v}^T$.Similarly, we have $J_{\boldsymbol{v}'}^{d_n} = -2\boldsymbol{v}'^T$. Now, (26) can be rewritten as

$$\Lambda_{d_n} = 4(\boldsymbol{v}^T \Lambda_{\boldsymbol{v}} \boldsymbol{v} + \boldsymbol{v}'^T \Lambda_{\boldsymbol{v}'} \boldsymbol{v}'). \tag{27}$$

Therefore, the norm constraint (21) is finally expressed in the real case as

$$\frac{d_n^2}{\Lambda_{d_n}} < \kappa_n, \tag{28}$$

where d_n is computed by (25), Λ_{d_n} by (27), and κ_n is a coefficient.

In fact, $\frac{d_n^2}{\Lambda_{d_n}}$ can be considered up to the first-order approximation that we have used as a random variable following a χ^2 distribution with 1 degree of freedom. Looking at the χ^2 table, we can choose an appropriate value for κ_n. This gives us the probability that d_n^2 falls into the interval $[0,\ \kappa_n \Lambda_{d_n}]$. For example, we can take $\kappa_n = 3.84$ for a probability of 95% when we consider the lengths of segments and $\kappa_n = 1.32$ for a probability of 75% when we consider the distance between the midpoints of the two segments, that is, we impose a stricter constraint on the distance between midpoints than on the lengths of segments.

B. Dot-Product Constraint

The dot-product constraint says that the difference between the cosines of angles between two vectors should be zero. Suppose we are given two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ and their covariance matrices $\Lambda_{\boldsymbol{u}}$ and $\Lambda_{\boldsymbol{v}}$. Here, $\boldsymbol{u}$ and $\boldsymbol{v}$ are assumed unit vectors. Denote the difference between the cosines of angles as d_c, i.e.

$$d_c = \boldsymbol{u} \cdot \boldsymbol{v} - \boldsymbol{u}' \cdot \boldsymbol{v}' = \boldsymbol{u}^T \boldsymbol{v} - \boldsymbol{u}'^T \boldsymbol{v}'. \tag{29}$$

We now compute the variance Λ_{d_c} of d_c. Under the first-order approximation, we have

$$\Lambda_{d_c} = J_{\boldsymbol{u}}^{d_c} \Lambda_{\boldsymbol{u}} J_{\boldsymbol{u}}^{d_c\,T} + J_{\boldsymbol{v}}^{d_c} \Lambda_{\boldsymbol{v}} J_{\boldsymbol{v}}^{d_c\,T} + J_{\boldsymbol{u}'}^{d_c} \Lambda_{\boldsymbol{u}'} J_{\boldsymbol{u}'}^{d_c\,T} + J_{\boldsymbol{v}'}^{d_c} \Lambda_{\boldsymbol{v}'} J_{\boldsymbol{v}'}^{d_c\,T} \tag{30}$$

where $J_{\boldsymbol{u}}^{d_c}$, $J_{\boldsymbol{v}}^{d_c}$, $J_{\boldsymbol{u}'}^{d_c}$ and $J_{\boldsymbol{v}'}^{d_c}$ are the Jacobian matrices of d_c with respect to $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{u}'$, and $\boldsymbol{v}'$, respectively. We assume that $\boldsymbol{u}$, $\boldsymbol{v}$, $\boldsymbol{u}'$, and $\boldsymbol{v}'$ are four independent Gaussian random variables.

The Jacobian $J_{\boldsymbol{u}}^{d_c}$ is given by $J_{\boldsymbol{u}}^{d_c} = \frac{\partial d_c}{\partial \boldsymbol{u}} = \boldsymbol{v}^T$. Similarly, we have $J_{\boldsymbol{v}}^{d_c} = \boldsymbol{u}^T$,$J_{\boldsymbol{u}'}^{d_c} = -\boldsymbol{v}'^T$, and $J_{\boldsymbol{v}'}^{d_c} = -\boldsymbol{u}'^T$. Equation (30) can be rewritten as

$$\Lambda_{d_c} = \boldsymbol{v}^T \Lambda_{\boldsymbol{u}} \boldsymbol{v} + \boldsymbol{u}^T \Lambda_{\boldsymbol{v}} \boldsymbol{u} + \boldsymbol{v}'^T \Lambda_{\boldsymbol{u}'} \boldsymbol{v}' + \boldsymbol{u}'^T \Lambda_{\boldsymbol{v}'} \boldsymbol{u}'. \tag{31}$$

Therefore, the dot-product constraint (22) is finally expressed in the real case as

$$\frac{d_c^2}{\Lambda_{d_c}} < \kappa_c \tag{32}$$

where d_c is computed by (29), Λ_{d_c} (31), and κ_c is a coefficient. We can choose $\kappa_c = 1.32$ for a probability of 75%.

C. Triple-Product Constraint

The same manipulation can be done for the mix-product constraint. Given six unit vectors $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{u}_3$, $\boldsymbol{u}_1'$, $\boldsymbol{u}_2'$, $\boldsymbol{u}_3'$, and their covariance matrices. Denote the difference between the two triple products by d_t, i.e.

$$d_t = \langle \boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3 \rangle - \langle \boldsymbol{u}_1', \boldsymbol{u}_2', \boldsymbol{u}_3' \rangle. \tag{33}$$

We can compute the covariance Λ_{d_t} of d_t under the first-order approximation. The mix-product constraint (23) is then expressed in the real case as

$$\frac{d_t^2}{\Lambda_{d_t}} < \kappa_t \tag{34}$$

where κ_t is a coefficient.

In our implementation, however, we do not compute the variance of the d_t because the computation is relatively expensive. Instead, we give a predefined threshold ε_t. If $|d_t| < \varepsilon_t$, then the mix-product constraint is considered satisfied; otherwise, it is not satisfied. $|d_t|$ reaches a maximum of 2 when $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, and $\boldsymbol{u}_3$ are perpendicular to each other, $\boldsymbol{u}_1'$, $\boldsymbol{u}_2'$, $\boldsymbol{u}_3'$ are perpendicular to each other, and they are not congruent (i.e., reflection). $|d_t|$ reaches a minimum of 0 when they are congruent. We have chosen $\varepsilon_t = 0.5$ in our implementation to account for noise.

VI. The Displacement Estimation Algorithm

In this section, we present an algorithm based on the hypothesis-and-verify paradigm that registers two stereo frames and estimates the displacement between them. We assume that the environment is static and that it is only the stereo rig that has moved.

A. Generating Hypotheses

At the first stage, the rigidity constraints are heavily used to generate hypotheses of matches between the two frames. Two pairings of segments form a plausible hypothesis if they satisfy the rigidity constraints described in the previous sections. If we explore all possible pairs, the complexity is $O(m^2n^2)$, where m is the number of segments in the first frame, and n is that in the second frame. The generation of hypotheses is implemented as follows. For a segment S_1 in the first frame, we find a segment S_1' in the second frame such that its length is compatible with that of S_1. For the pairing (S_1, S_1'), we then find the pairings (S_k, S_k') such that the two pairings form a plausible hypothesis. When we are done, we go to the next segment S_2 of the first frame.

Since at this stage we do not want to recover all matches between two frames, but to recover all potential displacements between them, we reduce the complexity of the hypothesis generation phase by using a number of heuristics:

Sort the segments: Sort all segments in each frame in decreasing length order so that we can easily find, by binary search, the segments in the second frame that are compatible in length with the segments in the first one.

Control the depth: Rather than finding all possible pairings compatible with a given pairing, we stop if we have found a sufficient number of compatible pairings (five, for instance).

Avoid redundant hypotheses: If a pairing has already been retained as a potential match in some earlier hypothesis, it is not considered further because it does not give us new information about the displacement between two frames.

Reduce the width: Consider segments of the first frame only in the central part of the frame because segments on the sides are likely to have moved out of the field of view in the next frame.

Reduce the number of segments: Choose only the longest segments in the first scene, for instance, the m/q longest (q = 2 or 3) . This also reduces the search width.

Other constraints can be integrated in the algorithm to increasingly speed up the generation process. In the indoor mobile robot navigation example, the robot and the objects usually move horizontally (in the ground plane). In our stereo coordinate system, the ground plane is parallel to the plane $y = 0$. Thus, we can impose the constraint that if two segments can be matched to each other, they must have almost the same y coordinates. Another constraint we can use is on the change of the orientation of a segment [47]. In general, the rotation angle between two successive frames does not go beyond 60°; therefore, we can impose that the orientation difference of a pairing of segments to be matched must be less than 60°.

B. Verifying Hypotheses

At the second stage of the algorithm, we propagate each hypothesis generated above to the whole frame and try to match more segments. We then choose the best hypotheses, where best is defined later. This can be done in parallel for each hypothesis. The process is performed as follows.

1) Estimating the Initial Rigid Displacement: From Proposition 4, we know that a unique rigid displacement can be computed for each hypothesis. We compute an initial estimation of the rotation and translation via an extended Kalman filter (see Section VII). An initial guess of the state vector (representing the rigid displacement) is required as input to the filter. In our implementation, the state vector $\boldsymbol{s}$ for the displacement is simply initialized to zero (i.e., $\hat{\boldsymbol{s}}_0 = 0$) but with a big covariance matrix. The covariance matrix $\Lambda_{\hat{\boldsymbol{s}}_0}$ is initialized as follows: $\Lambda_{\hat{\boldsymbol{s}}_0}[i][i] = 2.0$ for $i = 0, 1, 2$ (rotation components), $\Lambda_{\hat{\boldsymbol{s}}_0}[i][i] = 1.0 \times 10^6$ for $i = 3, 4, 5$ (translation components), and $\Lambda_{\hat{\boldsymbol{s}}_0}[i][j] = 0$ for $i \neq j$. This is equivalent to saying that we assume a standard deviation of 114° of rotation along each axis and a standard deviation of 1 m of translation along each axis.

Because we use a first-order approximation, if the initial estimate is not very good, the final estimate given by the filter may be different from the true value. In order to reduce the effect of nonlinearities, we can apply the Kalman filter iteratively. Usually, we obtain a good estimate after a few iterations (typically three or four).

2) Propagating Hypotheses: We now have an initial estimate of the displacement for each hypothesis. We apply this estimate to the first frame and compare the transformed frame with the second one. If a transformed segment from the first frame is near enough to some segment in the second frame, then this pair is considered to be matched (see Section VII for the definition of near enough), and again, the extended Kalman filter is used to update the displacement estimate. After all segments have been processed, we obtain an estimate of the displacement, its uncertainty, and the correspondences between segments.

We said that we transformed all segments of the first frame once, using the initial estimate of displacement. This is the one-shot approach. The matching results will depend heavily on this initial estimate. Since we may have a poor initial estimate, we may not get a satisfactory matching result. To overcome this problem, we transform only one nonmatched segment of the first frame at a time. This is the many-shots approach. If we find a match for it, we update the displacement estimate and transform another nonmatched segment of the first frame using the new updated estimation of the displacement (no longer the initial estimate). If we cannot find a match, we process another segment of the first frame. After all segments have been processed, we obtain an estimate of the displacement, its uncertainty, and the correspondences between segments.

We have implemented the two approaches and found that the second one gives a much better performance. Starting from two slightly different initial estimates, the first approach yields two different (but not dramatically different, of course) results of matchings and displacement estimates, whereas the second converges to almost the same result.

In order to obtain a precise estimation and because of the nonlinearities, we iterate the above procedure twice.

3) Complexity: We note that the complexity of the algorithm in the worst case is $O(mn)$ for each hypothesis. The speed of the algorithm depends essentially on the ability to quickly access the segments of Frame 2 that are close to a transformed segment $\hat{S}$. We have used several approaches to

achieve this. One approach is to use binary search to discard segments of Frame 2 that are not compatible in length with S. Another efficient method is to use *bucketing techniques*, which allow us to obtain a list of segments in the neighborhood of some segment. The preprocessing necessary to sort segments into buckets can be done very quickly (the complexity is linear in the number of segments).

4) Choosing the Best Hypothesis: We now discuss how to choose the best hypothesis using the error given by the Kalman filter. The criterion must be a function of the number of segments actually matched and of the error made in approximating the match by a rigid displacement. We use the following criterion:

$$C = \sum_{i=1}^{p} E_i + (N - p)E_{min} \tag{35}$$

where E_i is the error of the ith match (the sum of the distances given by (52) and (53); see Section VIII), $N = \min(m, n)$ is the smallest number of segments in the two scenes, p is the number of segments matched, and E_{min} is the error corresponding to the threshold determining when two segments are matched (which is equal to $\kappa_{\boldsymbol{\Psi}} + \kappa_{\boldsymbol{m}}$, which is given in Section VIII). Then, the hypothesis with a minimal C is chosen as the best one. From (35), we can see that if we have more matches, the error is smaller. Note that if we simply define C as p (the number of segments matched) and choose the hypothesis with a maximal C as the best one, almost the same result is obtained.

VII. Estimating 3-D Displacement via an Extended Kalman Filter

Given a set of correspondences of 3-D primitives (points, line segments, and planar patches), many methods are reported in the literature to estimate 3-D displacement [35], [36], [38]–[40], [56]. A comparison of several of those methods has been carried out using 3-D line segment correspondences (see [57]). This work reveals that the extended Kalman filter should be preferred for its efficiency and accuracy. We assume that the reader is familiar with the ideas and equations of the Kalman filter [58], [59], which have now become of standard use in vision.

The standard Kalman filter is a powerful tool to deal with parameter estimation problems in a linear noisy system. However, it is not directly applicable to our problem because of its nonlinearity. The extended Kalman filter approach [58], [60] applies the standard Kalman filter to *nonlinear* systems with additive Gaussian noise by continually updating a *linearization* around the previous state estimate, starting with an initial guess. Here, we formulate the displacement estimation problem using the representation of 3-D line segments described in Section IV in order to apply the extended Kalman filter. We first briefly describe our representation of rotations.

Many representations exist for 3-D rotations, including Euler angles, quaternions, and rotation axes [61], [62]. We use the rotation axis representation. More precisely, a rotation can be defined as a 3-D vector $\boldsymbol{r} = [a, b, c]^T$, whose direction is that of the axis of rotation and whose magnitude is equal to the rotation angle. The relation between $\boldsymbol{r}$ and the rotation matrix $\boldsymbol{R}$ is given by the following formula, which is known as the Rodrigues formula [63]:

$$\boldsymbol{R} = \boldsymbol{I}_3 + \frac{\sin\theta}{\theta}\tilde{\boldsymbol{r}} + \frac{1 - \cos\theta}{\theta^2}\tilde{\boldsymbol{r}}^2 \tag{36}$$

where θ is the norm of $\boldsymbol{r}$ (i.e., $\theta = \|\boldsymbol{r}\|$), $\boldsymbol{I}_3$ is the 3×3 identity matrix, and $\tilde{\boldsymbol{r}}$ is the antisymmetric matrix defined as

$$\tilde{\boldsymbol{r}} = \begin{bmatrix} 0 & -c & b \\ c & 0 & -a \\ -b & a & 0 \end{bmatrix}.$$

This matrix represents the crossproduct with vector $\boldsymbol{r}$ since $\tilde{\boldsymbol{r}}\boldsymbol{x} = \boldsymbol{r} \wedge \boldsymbol{x}$ for all x. A 3-D displacement is then represented by the 6-D vector $\boldsymbol{s} = [\boldsymbol{r}^T, \boldsymbol{t}^T]^T$.

For reason of simplicity, we define two nonlinear functions $\boldsymbol{g}$ and $\boldsymbol{h}$ to relate a unit vector $\boldsymbol{u}$ and its Euler angles $\boldsymbol{\Psi}$ (see (8) and (11)) such that

$$\boldsymbol{\Psi} = \boldsymbol{g}(\boldsymbol{u}) \quad \text{and} \quad \boldsymbol{u} = \boldsymbol{h}(\boldsymbol{\Psi}). \tag{37}$$

If a segment S is characterized by $(\boldsymbol{\Psi}_1, \boldsymbol{m}_1)$ in Frame 1 and by $(\boldsymbol{\Psi}_2, \boldsymbol{m}_2)$ in Frame 2 and the displacement between the two frames is $\boldsymbol{s} = [\boldsymbol{r}^T, \boldsymbol{t}^T]^T$, then we have the following equations:

$$\boldsymbol{\Psi}_2 = \boldsymbol{g}(\boldsymbol{R}\boldsymbol{h}(\boldsymbol{\Psi}_1)), \tag{38}$$

$$\boldsymbol{m}_2 = \boldsymbol{R}\boldsymbol{m}_1 + \boldsymbol{t}. \tag{39}$$

The above equations simply say that the *transformed* segment of the first frame should have the same orientation and location as the segment observed in the second frame. If we define the measurement vector as

$$\boldsymbol{x} = [\boldsymbol{\Psi}_1^T, \boldsymbol{m}_1^T, \boldsymbol{\Psi}_2^T, \boldsymbol{m}_2^T]^T \tag{40}$$

then we can write down the measurement equation from (38) and (39) as

$$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s}) = \begin{bmatrix} \boldsymbol{g}(\boldsymbol{R}\boldsymbol{h}(\boldsymbol{\Psi}_1)) - \boldsymbol{\Psi}_2 \\ \boldsymbol{R}\boldsymbol{m}_1 + \boldsymbol{t} - \boldsymbol{m}_2 \end{bmatrix} = \boldsymbol{0}. \tag{41}$$

This is a 5-D vector equation. In the following, the first two components of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s})$ are denoted by $\boldsymbol{f}_1$ and the last three components by $\boldsymbol{f}_2$.

Equation (41) can be expanded into a Taylor series around the current displacement estimate $\hat{\boldsymbol{s}}$ and the current observation $\hat{\boldsymbol{x}}$:

$$\begin{aligned} \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s}) =& \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}}) + \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})}{\partial \boldsymbol{x}}(\boldsymbol{x} - \hat{\boldsymbol{x}}) + \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})}{\partial \boldsymbol{s}}(\boldsymbol{s} - \hat{\boldsymbol{s}}) \\ &+ O((\boldsymbol{x} - \hat{\boldsymbol{x}})^2) + O((\boldsymbol{s} - \hat{\boldsymbol{s}})^2). \end{aligned} \tag{42}$$

Ignoring the second-order terms, we get a linearized measurement equation:

$$\boldsymbol{y} = \boldsymbol{M}\boldsymbol{s} + \boldsymbol{v} \tag{43}$$

where $\boldsymbol{y}$ is the new measurement vector, $\boldsymbol{v}$ is the noise vector of the new measurement, and $\boldsymbol{M}$ is the linearized transformation

matrix. They are given by

$$\boldsymbol{M} = \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})}{\partial \boldsymbol{s}},$$
$$\boldsymbol{y} = -\boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}}) + \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})}{\partial \boldsymbol{s}}\hat{\boldsymbol{s}},$$
$$\boldsymbol{v} = \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})}{\partial \boldsymbol{x}}(\boldsymbol{x} - \hat{\boldsymbol{x}}),$$

with $E[\boldsymbol{v}] = \mathbf{o}$, $E[\boldsymbol{v}\boldsymbol{v}^T] = \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})}{\partial \boldsymbol{x}} \Lambda_{\hat{\boldsymbol{x}}} \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{x}}, \hat{\boldsymbol{s}})}{\partial \boldsymbol{x}}^T$.

The standard Kalman filter can then be applied to the above linearized system to update the displacement estimate $\hat{\boldsymbol{s}}$. The derivatives of $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{s})$ with respect to $\boldsymbol{s}$ and to $\boldsymbol{x}$ are computed as follows:

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{s}} = \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{t}} & \mathbf{o} \\ \frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{r}} & \boldsymbol{I}_3 \end{bmatrix} \tag{44}$$

$$\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} = \begin{bmatrix} \frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\Psi}_1} & \mathbf{o} & -\boldsymbol{I}_2 & \mathbf{o} \\ \mathbf{o} & \boldsymbol{R} & \mathbf{o} & -\boldsymbol{I}_3 \end{bmatrix} \tag{45}$$

where

$$\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{r}} = \frac{\partial \boldsymbol{g}(\boldsymbol{u}_1')}{\partial \boldsymbol{u}} \frac{\partial (\boldsymbol{R}\boldsymbol{u}_1)}{\partial \boldsymbol{r}},$$
$$\frac{\partial \boldsymbol{f}_2}{\partial \boldsymbol{r}} = \frac{\partial (\boldsymbol{R}\boldsymbol{m}_1)}{\partial \boldsymbol{r}}$$
$$\frac{\partial \boldsymbol{f}_1}{\partial \boldsymbol{\Psi}_1} = \frac{\partial \boldsymbol{g}(\boldsymbol{u}_1')}{\partial \boldsymbol{u}} \boldsymbol{R} \frac{\partial \boldsymbol{h}(\boldsymbol{\Psi}_1)}{\partial \boldsymbol{\Psi}}$$

with $\boldsymbol{u}_1 = \boldsymbol{h}(\boldsymbol{\Psi}_1)$ and $\boldsymbol{u}_1' = \boldsymbol{R}\boldsymbol{u}_1$. The reader is referred to [48] for the computation of the derivatives $\frac{\partial (\boldsymbol{R}\boldsymbol{u}_1)}{\partial \boldsymbol{r}}$, $\frac{\partial (\boldsymbol{R}\boldsymbol{m}_1)}{\partial \boldsymbol{r}}$, $\frac{\partial \boldsymbol{g}(\boldsymbol{u}_1')}{\partial \boldsymbol{u}}$, and $\frac{\partial \boldsymbol{h}(\boldsymbol{\Psi}_1)}{\partial \boldsymbol{\Psi}}$.

Each time a new match becomes available, we can incrementally update the displacement estimate $\boldsymbol{s}$ via extended Kalman filtering using the above formulation.

VIII. Matching Noisy Line Segments

Suppose now that we have an estimate of the displacement $\boldsymbol{s} = [\boldsymbol{r}^T, \boldsymbol{t}^T]^T$ between the two frames with its covariance matrix $\Lambda_{\boldsymbol{s}}$. The question is the following: for a given segment S in Frame 1, which segment in Frame 2 can it be matched to? Our solution has two steps. The segment S is first transformed into $\widehat{S}$ by applying the given displacement estimate. It is then compared with segments in Frame 2.

A. Transformation of a 3-D Line Segment

Let the parametrization of the segment S be $(\boldsymbol{\Psi}, \boldsymbol{m})$ and that of the transformed segment $\widehat{S}$ be $(\widehat{\boldsymbol{\Psi}}, \widehat{\boldsymbol{m}})$. Clearly, we have

$$\widehat{\boldsymbol{\Psi}} = \boldsymbol{g}(\boldsymbol{R}\boldsymbol{h}(\boldsymbol{\Psi})), \tag{46}$$
$$\widehat{\boldsymbol{m}} = \boldsymbol{R}\boldsymbol{m} + \boldsymbol{t}. \tag{47}$$

If the uncertainty measure of the segment S is given by $(\Lambda_{\boldsymbol{\Psi}}, \Lambda_{\boldsymbol{m}})$, then we can compute the covariance matrix $\Lambda_{\widehat{\boldsymbol{\Psi}}}$ of $\widehat{\boldsymbol{\Psi}}$, under the first-order approximation, as

$$\Lambda_{\widehat{\boldsymbol{\Psi}}} = J_{\boldsymbol{\Psi}}^{\widehat{\boldsymbol{\Psi}}} \Lambda_{\boldsymbol{\Psi}} J_{\boldsymbol{\Psi}}^{\widehat{\boldsymbol{\Psi}}^T} + J_{\boldsymbol{s}}^{\widehat{\boldsymbol{\Psi}}} \Lambda_{\boldsymbol{s}} J_{\boldsymbol{s}}^{\widehat{\boldsymbol{\Psi}}^T}, \tag{48}$$

where the Jacobian matrices are given by

$$J_{\boldsymbol{\Psi}}^{\widehat{\boldsymbol{\Psi}}} = \frac{\partial \boldsymbol{g}(\hat{\boldsymbol{u}})}{\partial \boldsymbol{u}} \boldsymbol{R} \frac{\partial \boldsymbol{h}(\boldsymbol{\Psi})}{\partial \boldsymbol{\Psi}},$$
$$J_{\boldsymbol{s}}^{\widehat{\boldsymbol{\Psi}}} = \begin{bmatrix} \frac{\partial \boldsymbol{g}(\hat{\boldsymbol{u}})}{\partial \boldsymbol{u}} \frac{\partial (\boldsymbol{R}\boldsymbol{u})}{\partial \boldsymbol{r}} & \mathbf{o}_3 \end{bmatrix}$$

with $\boldsymbol{u} = \boldsymbol{h}(\boldsymbol{\Psi})$, $\hat{\boldsymbol{u}} = \boldsymbol{R}\boldsymbol{u}$, and $\mathbf{o}_3$ is the 3×3 null matrix.

Similarly, the covariance matrix $\Lambda_{\widehat{\boldsymbol{m}}}$ of the midpoint $\widehat{\boldsymbol{m}}$ is given by

$$\Lambda_{\widehat{\boldsymbol{m}}} = \boldsymbol{R}\Lambda_{\boldsymbol{m}}\boldsymbol{R}^T + J_{\boldsymbol{s}}^{\widehat{\boldsymbol{m}}} \Lambda_{\boldsymbol{s}} J_{\boldsymbol{s}}^{\widehat{\boldsymbol{m}}^T} \tag{49}$$

where the Jacobian matrix of $\widehat{\boldsymbol{m}}$ with respect to $\boldsymbol{s}$ is

$$J_{\boldsymbol{s}}^{\widehat{\boldsymbol{m}}} = \begin{bmatrix} \frac{\partial (\boldsymbol{R}\boldsymbol{m})}{\partial \boldsymbol{r}} & \boldsymbol{I}_3 \end{bmatrix}.$$

The reader is referred to [48] for the computation of the derivatives.

B. Criterion of Matching

Now, we have a representation of the segment $\widehat{S}$ and a set of segments $\{S_i | i = 1..n\}$ in Frame 2. We want to decide which segment S_i can be matched with $\widehat{S}$.

Using the representation of 3-D line segments described in Section IV, matching can be done very efficiently. It goes in two steps. Let the segment S_i be represented by $(\boldsymbol{\Psi}_i, \boldsymbol{m}_i)$ and its covariance matrices $(\Lambda_{\boldsymbol{\Psi}_i}, \Lambda_{\boldsymbol{m}_i})$. We want to know whether S_i can be matched with $\widehat{S}$ or not. The first step is to examine the similarity in orientation. The Mahalanobis distance between the orientations is given by

$$d_{\boldsymbol{\Psi}} = (\boldsymbol{\Psi} - \boldsymbol{\Psi}_i)^T (\Lambda_{\boldsymbol{\Psi}} + \Lambda_{\boldsymbol{\Psi}_i})^{-1} (\boldsymbol{\Psi} - \boldsymbol{\Psi}_i). \tag{50}$$

If $d_{\boldsymbol{\Psi}}$ is less than some threshold $\kappa_{\boldsymbol{\Psi}}$, we then go to the second step: examine the distance between the midpoints of two segments. The Mahalanobis distance between the midpoints is given by

$$d_{\boldsymbol{m}} = (\boldsymbol{m} - \boldsymbol{m}_i)^T (\Lambda_{\boldsymbol{m}} + \Lambda_{\boldsymbol{m}_i})^{-1} (\boldsymbol{m} - \boldsymbol{m}_i). \tag{51}$$

If $d_{\boldsymbol{m}}$ is still less than some threshold $\kappa_{\boldsymbol{m}}$, we then consider the two segments to be matched.

Before computing the Mahalanobis distance between the $\boldsymbol{\Psi}$'s (52), special care is required to cope with the discontinuity of Ψ when a segment is nearly parallel to the plane $y = 0$ (see Section IV). The treatment is very simple: If $\Psi < \pi/2$ and $\Psi_i > 3\pi/2$, then set $\Psi_i = \Psi_i - 2\pi$; else, if $\Psi > 3\pi/2$ and $\Psi_i < \pi/2$, then set $\Psi = \Psi - 2\pi$; else, do nothing. Note that adding a constant to a random variable does not affect its covariance matrix.

By appropriately choosing the threshold $\kappa_{\boldsymbol{\Psi}}$ and $\kappa_{\boldsymbol{m}}$, we can recover almost all possible matches. From the χ^2 distribution table, we take $\kappa_{\boldsymbol{\Psi}} = 6.0$ for a probability of 95% with 2 degrees of freedom, and $\kappa_{\boldsymbol{m}} = 7.8$ for a probability of 95% with 3 degrees of freedom.

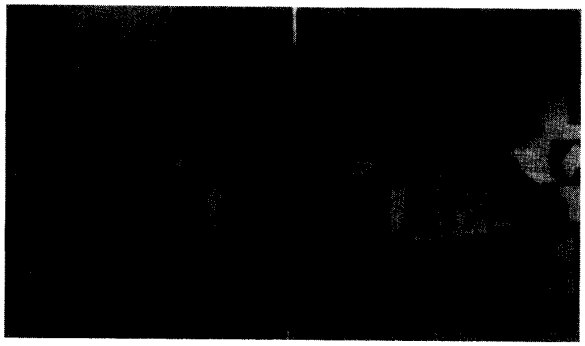


Fig. 4. Images of the first camera: The left one is at t_1, and the right one is at t_2.

IX. Experimental Results

The algorithm presented in this paper has been tested for two years using real stereo data acquired by our trinocular stereo system. At least 200 pairs of stereo frames (i.e., 400 individual stereo frames) have been used. The algorithm has succeeded in correctly matching and computing the 3-D displacement and matching two views in almost all cases. The few cases of failure are due to too little common intersection (less than 20%) between the two stereo frames.

The reader can find some experimental results in [47]. In this section, we provide two examples to demonstrate the matching process. In each figure shown, if there are four pictures, the upper left one is the top view (projection on the ground plane), the lower left one is the front view (projection on a plane in front of the stereo system and perpendicular to the ground plane), the lower right one is a side view (projection on the plane perpendicular to the previous two planes), and the upper right is the view from the first camera (perspective projection using the calibrated camera transformation matrix). If there are only two pictures in each figure, then the left one is the front view, and the right one is the top view.

A. Indoor Scene

Fig. 4 shows the images of an indoor scene taken by the first camera of the stereo rig at two instants. As can be observed, there is only a small common part between two successive frames. In fact, there is a big rotation between the two views. Comparing the two images in Fig. 4, the boxes on the table have a displacement of about 200 pixels in the image plane (resolution: 512×512).

Figs. 5 and 6 show the pair of stereo frames reconstructed by our trinocular stereo. The triangle in each picture represents the optical centers of the three cameras. We have 79 segments in the first frame and 121 segments in the second. There is a rotation of about 16.5° between the two positions, which can be noted by superposing the two frames (see Fig. 7).

Applying the displacement estimation algorithm to these two frames, we obtain five hypotheses. All five hypotheses are propagated to the whole frame to match more segments and to update the displacement estimate. In the end, four hypotheses yield the correct estimate of the displacement.

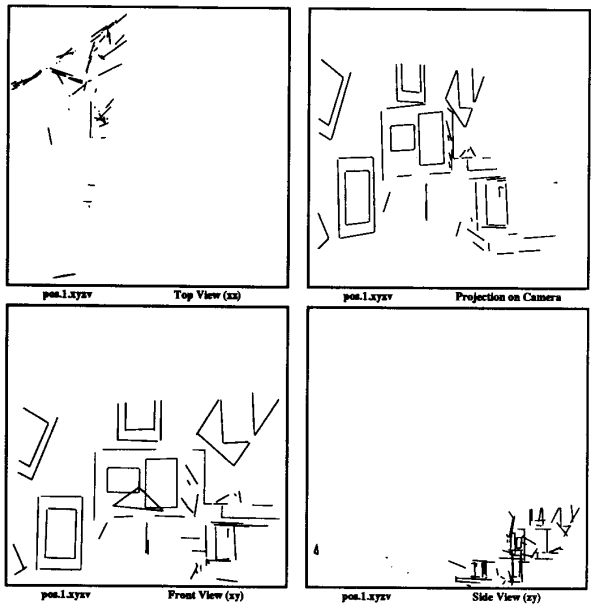


Fig. 5. Different views of the first 3-D frame (uniform scale).

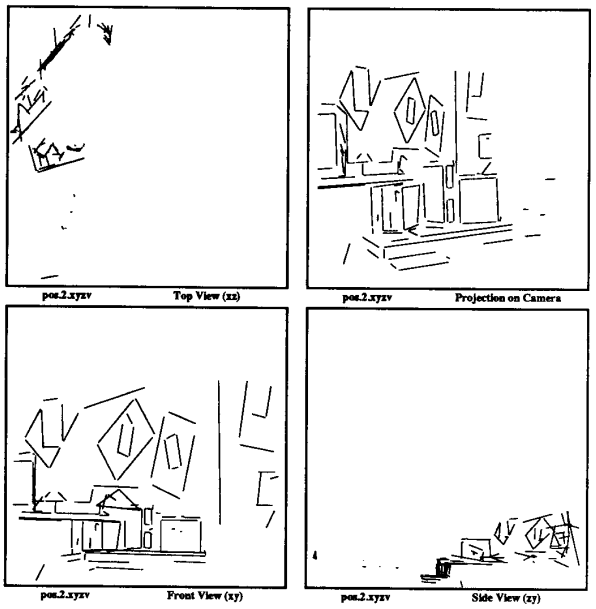


Fig. 6. Different views of the second 3-D frame (uniform scale).

Thirty seven matches are recovered. To determine how good this estimate is, we apply the computed estimate to the first frame and superimpose the transformed one on the second. Fig. 8 shows such superposition of the matched segments, and Fig. 9 shows the superposition of the whole frames. The estimate of displacement is very good. We can observe the rotation between the two positions by the shift of the two triangles in Fig. 9. We observe also that the common part is very small. The whole process takes about 50 s on a SUN 3 workstation.

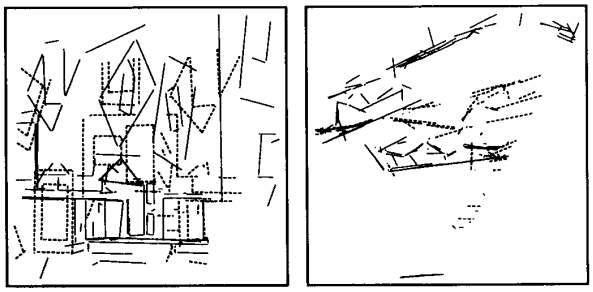


Fig. 7. Superposition of the two stereo frames: Segments of Frame 1 are represented in dashed lines and those of Frame 2 in solid lines.

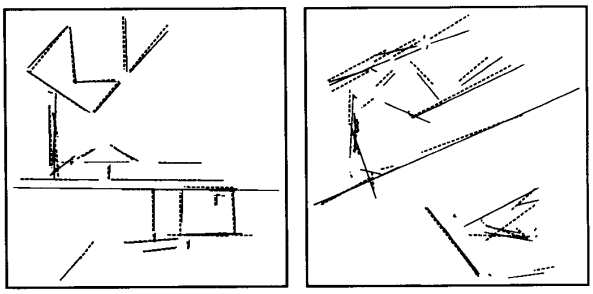


Fig. 8. Superposition of the matched segments after applying the computed displacement to the segments of Frame 1: Segments of Frame 1 are represented by dashed lines and those of Frame 2 in solid lines.

Fig. 9. Superposition of the transformed segments of Frame 1 (in dashed lines) and those of Frame 2 (in solid lines) (nonuniform scale).

Several remarks can be made at this point:

- Segments in the foreground (near the observer) are superimposed better than others. This is reasonable since segments close to the observer are more precisely reconstructed by the stereo system than distant ones.

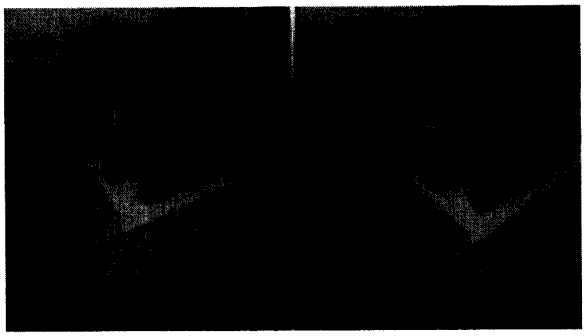


Fig. 10. Two images of a rock scene taken by the first camera.

- There is a better agreement in the lateral coordinates of the segments than in the range because the range component is usually much more uncertain than the other components.
- Those two remarks justify the use of the weighted least-squares property of the Kalman filtering approach.

B. Rock Scene

As pointed out earlier, our program is developed in the context of visual navigation of a mobile robot in an indoor scene. We expect to be able to describe most of the objects in such an environment by line segments. In this subsection, we describe an example that shows that our algorithm works in a cluttered scene containing rocks. Fig. 10 shows two images of such a rock scene taken by the first camera of the stereo rig at two instants.

Two 3-D frames are reconstructed by our trinocular stereo system displayed in Figs. 11 and 12. We have obtained 211 3-D line segments in the first frame and 208 in the second. Two remarks can be made. The first is that the segments reconstructed are very noisy and even spurious (for example, some are under the ground, as can be observed in the front views); the second is that they are very small. As we can observe, there is a rather big difference between the two frames (about 11° of rotation and 34 cm of translation).

In our program, a parameter l, ranging from -5 to 5, is used to control the thresholds in the rigidity constraints. A change of one for l corresponds to a change of 0.2 for the thresholds. For example, the threshold for the dot-product constraint $\kappa_c = 1.32 + 0.2l$. The parameter l is set to zero in the program, which corresponds to the thresholds indicated in Section V. When l decreases, the constraints are imposed more strictly during the hypothesis generation process. When l increases, the constraints are imposed less strictly. For indoor scenes, we do not need to change l, i.e., the thresholds are image independent (they are certainly stereo-system dependent). However, as the 3-D frames reconstructed from the rock scene are very noisy, the motion program generates many false hypotheses, although the final result is good. If we set l to -3, i.e., we impose more strict rigidity constraints, we obtain five hypotheses. All these hypotheses are propagated to the whole frame to match more segments and

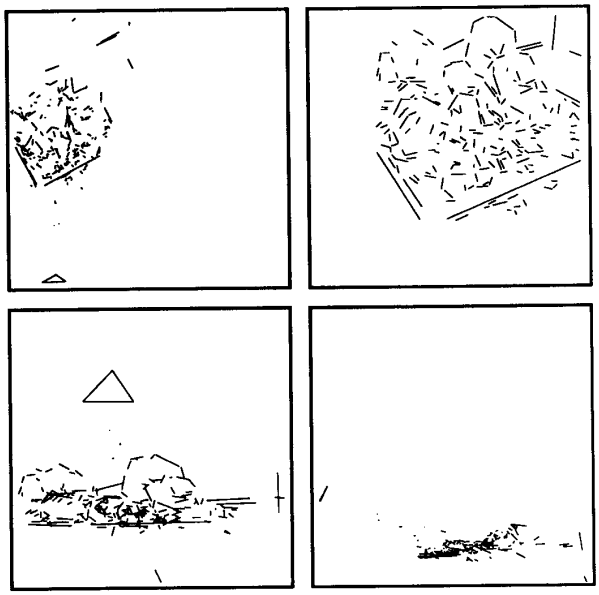


Fig. 11. Different views of the 3-D frame reconstructed at t_1 (uniform scale).

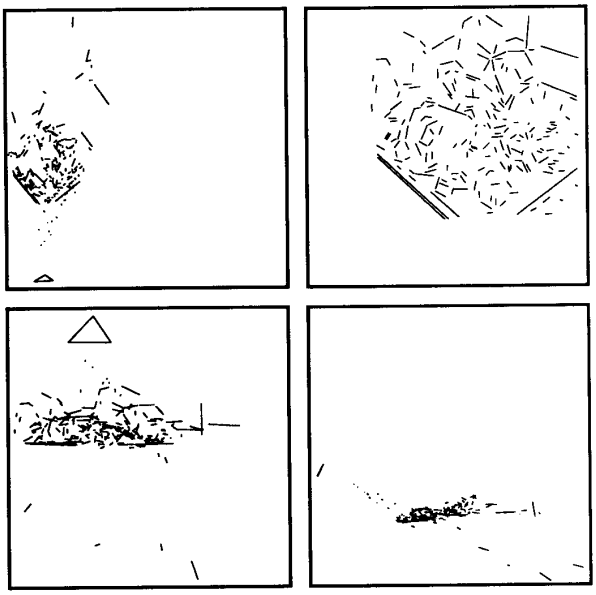


Fig. 12. Different views of the 3-D frame reconstructed at t_2 (uniform scale).

to update the motion estimate. In the end, four hypotheses give the correct estimate of the displacement. Ninety three matches are recovered. To determine how good this estimate is, we apply the computed estimate to the first frame and superimpose the transformed one on the second, which is displayed in Fig. 13. We can find that the motion estimate is still very good even for such a complicated scene. The displacement of the robot is shown by the shift of the two triangles in the top view of the superposition. The fifth hypothesis gives a suboptimal solution (21 matches are found by this hypothesis). The whole

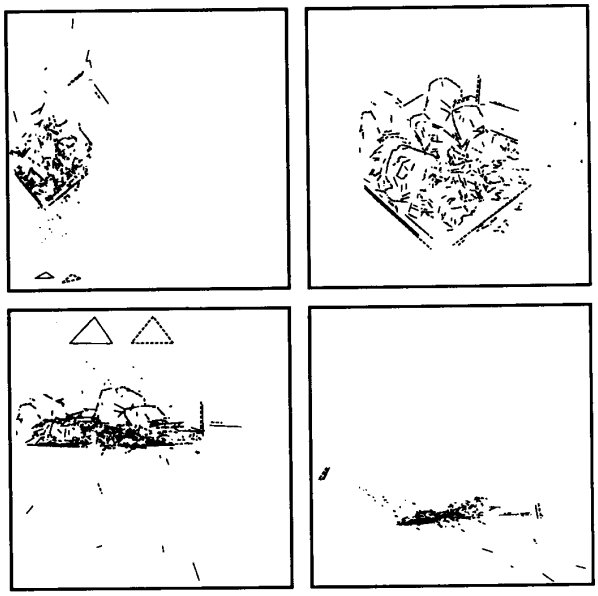


Fig. 13. Superposition of the transformed segments of Frame 1 (in dashed lines) and those of Frame 2 (in solid lines) (uniform scale).

process takes about 50 s on a SUN 4 workstation: 40 s in the hypothesis generation phase and 10 s in the verification phase.

X. Application to the Estimation of Multiple Object Displacements

Usually, in a static environment (an indoor scene, for example), the observer moves and at the same time there are other moving objects, or the observer is fixed but more than one moving objects exist (surveillance application, for example). We call this the *multiple object displacement* problem.

The algorithm described above is directly applicable to analyze the multiple object displacement problem. In the hypothesis generation phase, the algorithm tries to find all pairs of pairings of segments between two frames that satisfy the rigidity constraints. Each such pair is a hypothesis of potential displacement between two frames. If two segments in a hypothesis are from a single object, the hypothesis will give the displacement of that object. If an object is made of more than two segments and two of them are precisely reconstructed by the stereo system, then at least one hypothesis among all of them generated by the algorithm belongs to the object. "Precisely" is, of course, related to the thresholds in the rigidity constraints.

We first apply the algorithm of hypothesis generation to the two observed frames. In order not to miss any potential displacements between them, one modification should be made. Remember that in order to speed up the generation process, we choose only the m/q ($q = 2$ or 3) longest segments in the first frame. It is very likely that for a small object, none of its segments are among the m/q longest; thus, the displacement of such an object may not be detected. In order to avoid this, we use all segments in the first frame, i.e., choose $q = 1$. Once the hypotheses are generated, we apply the algorithm

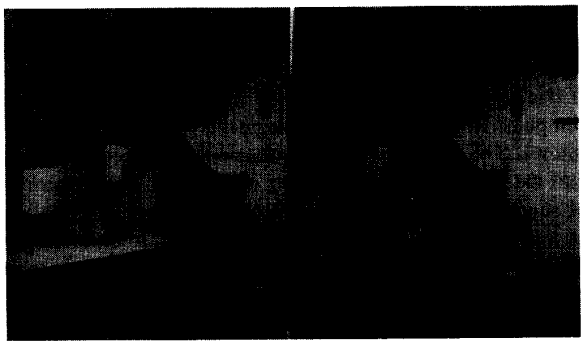


Fig. 14 Images taken by the first camera of the trinocular stereo rig: The left one is in the first position, and the right one is in the second position.

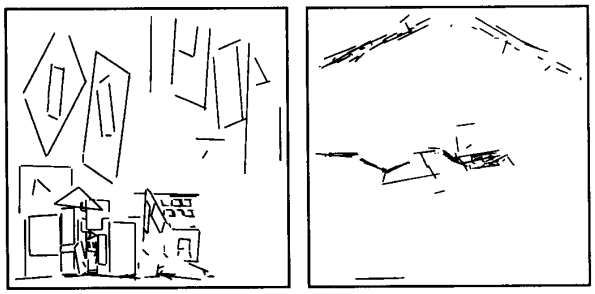


Fig. 15. Front and top views of the reconstructed 3-D frame in the first position.

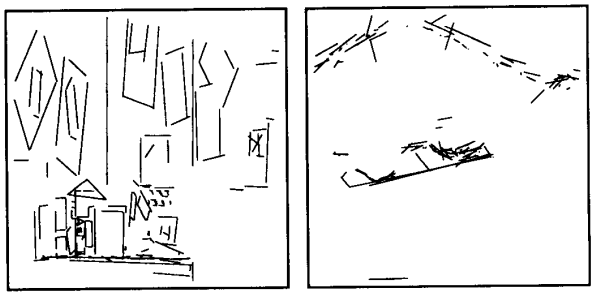


Fig. 16. Front and top views of the reconstructed 3-D frame in the second position.

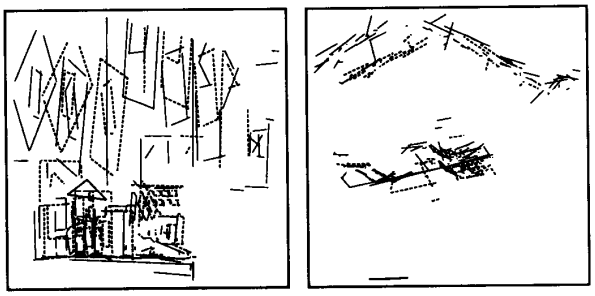


Fig. 17. Superposition of the two original frames.

of hypothesis verification to each of them to update the displacement estimate and to find segment correspondences. We then use the criterion described in (35) to sort those hypotheses. The best one is chosen as the displacement of Object 1, and the corresponding segments in the two frames are labeled as belonging to that object. For each of the other hypotheses, if they are not compatible with any previous one, we retain it as representing a new displacement. In the end, all displacements are recovered, and the scene is segmented into objects.

Two methods can be used to test the compatibility of two hypotheses. The first one is to compute the similarity of the corresponding displacement estimates. If the first hypothesis yields a displacement estimate $\boldsymbol{s}_1$ with its covariance matrix Λ_1 and the second one yields a displacement estimate $\boldsymbol{s}_2$ with its covariance matrix Λ_2, then the Mahalanobis distance between them can be computed as

$$d_{\boldsymbol{s}} = (\boldsymbol{s}_1 - \boldsymbol{s}_2)^T(\Lambda_1 + \Lambda_2)^{-1}(\boldsymbol{s}_1 - \boldsymbol{s}_2).$$

The distance $d_{\boldsymbol{s}}$ follows a χ^2 distribution with 6 degrees of freedom. We can choose $\kappa_{\boldsymbol{s}} = 12.6$ for a probability of 95%. If the distance between two hypotheses $d_{\boldsymbol{s}} \leq \kappa_{\boldsymbol{s}}$, they are considered compatible. Another approach is to look at the segment correspondences recovered by the hypotheses. For example, if half of the correspondences recovered by a hypothesis are among those recovered by another hypothesis, then the two hypotheses are considered to be compatible.

In [47], we have described an example of multiple object displacements using synthetic data. There, we also discussed the influence of the egomotion (the displacement of the robot) on the observed object displacements. The example below shows an experiment with a real scene with a real moving object. Fig. 14 shows two successive images observed by the first camera of the trinocular stereo rig. Notice that the moving object in the scene is the box in the foreground. It moves from right to left and toward the stereo rig that moves away from the observer. Figs. 15 and 16 display the reconstructed 3-D scenes in the two positions. Fig. 17 shows the overlay of the two 3-D frames and the difference between them. There are 168 segments in Frame 1 and 172 segments in Frame 2.

Considering that the static environment contains much more segments than the moving object, we first recover only the egomotion of the stereo rig. Only half of the longest segments in Frame 1 are used in the hypothesis generation process. Sixteen hypotheses are generated, and all of them are evaluated. Eight hypotheses give the estimation of the correct egomotion. Fig. 18 shows the estimated egomotion. The shift of the triangle indicates the displacement of the stereo rig. We remark that the two frames are well superimposed, except for the box. From the top view in Fig. 18, we easily observe the displacement of the box as indicated by arrows. In fact, five among the 16 hypotheses give the estimation of its displacement, but we do not intend to recover it at this stage because it may be deteriorated by some occasional bad alignment of segments belonging to the static environment [48].

There remain some unmatched segments. We apply the estimated egomotion to the first frame and remove the matched segments in the two frames. We thus obtain two 3-D frames containing only the segments of the moving object (and several

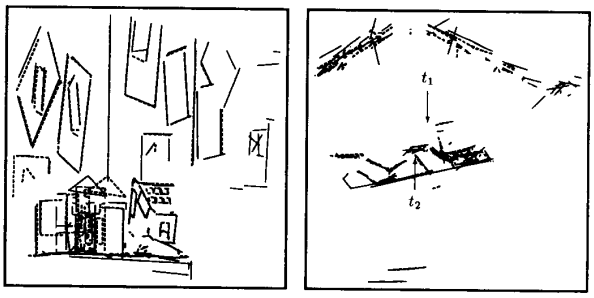


Fig. 18. Superposition of the segments of Frame 1 after applying the estimated egomotion and those of Frame 2.

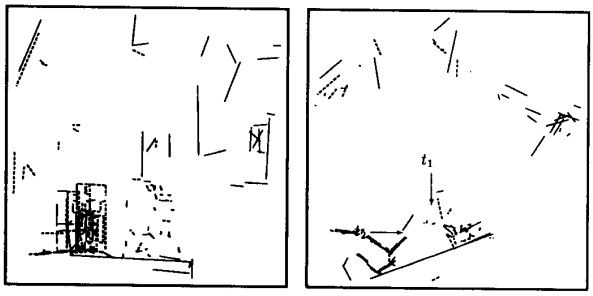


Fig. 19. Superposition of the remaining segments of both frames after those matched through the egomotion estimation procedure have been eliminated.

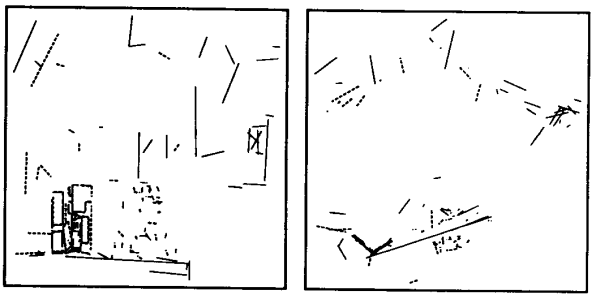


Fig. 20. Result of applying the estimated displacement of the box.

unmatched segments of the environment); the effect of the egomotion has been eliminated. The superposition of two such frames is displayed in Fig. 19. This figure clearly shows the movement of the box as indicated by arrows.

We then apply the same process as before to these two sets of segments. All segments are used in the hypothesis generation phase. Six hypotheses are generated, and they are all evaluated. All yield the correct displacement of the box. The result is shown in Fig. 20. We can remark that the segments of the box superimpose very well.

XI. Conclusion

We have developed an approach based on the *hypothesize-and-verify* paradigm for registering two stereo frames and computing the 3-D displacement between them. The rigidity constraints are used to generate hypotheses of feature correspondences between two frames. We have shown that the rigidity constraints we have formulated are complete for 3-D line segments and that a unique rigid displacement can be computed from two pairings of segments satisfying those constraints. The uncertainty of measurements has been integrated into the formalism of the rigidity constraints. If two pairings of segments satisfy the rigidity constraint, they are retained as a hypothesis. An initial estimate of the displacement can then be computed for each hypothesis. This initial estimate is propagated to the whole frame in an attempt to match more segments. Each time a new match is obtained, the displacement estimate is updated. Finally, the best hypothesis is retained. This algorithm has been successfully tested with several hundreds of real stereo frames.

We have also extended it to determine multiple object displacements. When the robot navigates in an environment in which there exist other moving rigid objects, our algorithm first determines the egomotion and then cancels it before recovering the object displacements.

The proposed algorithm is easily adapted to solve simple object recognition and location problems (see [48] for more details). It has many other interesting applications, such as visually guided navigation [64] and world model building [54].

Although the proposed algorithm has been successfully tested using a large number of scenes, the primitive used (3-D line segment) is rather limited. It is usually not sufficient to interpret complex scenes with only line segments. Our current research consists of developing strategies to include other primitives such as points, curves, and surface patches.

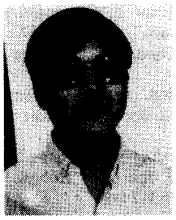
Acknowledgment

The authors would like to thank N. Ayache for his contribution to the work presented in this paper and the anonymous reviewers of the paper for their detailed comments and suggestions.

References

[1] T. Huang, "Motion analysis," in *AI Encyclopedia*. New York: Wiley, 1986, pp. 620–632.
[2] H. Nagel, "Image sequences—Ten (octal) years from phenomenology toward a theoretical foundation," in *Proc. 8th Int'l Conf. Patt. Recog.* (Paris, France), Oct. 1986, pp. 1174–1185.
[3] J. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images—A review," *Proc. IEEE*, vol. 76, pp. 917–935, Aug. 1988.
[4] S. T. Barnard and M. A. Fishler, "Computational stereo," *ACM Comput. Surveys*, vol. 14, pp. 553–572, Dec. 1982.
[5] W. Grimson, "Computational experiments with a feature based stereo algorithm," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, no. 1, pp. 17–34, 1985.
[6] S. Pollard, J. Mayhew, and J. Frisby, "PMF: A stereo correspondence algorithm using a disparity gradient limit," *Perception*, vol. 14, pp. 449–470, 1985.
[7] N. Ayache and B. Faverjon, "A fast stereo vision matcher based on prediction and recursive verification of hypotheses," in *Proc. IEEE Workshop Comput. Vision: Representation Contr.* Oct. 1985, pp. 27–37; Shorter version in *Proc. CVPR* (San Francisco, CA), 1985.
[8] N. Ayache and F. Lustman, "Fast and reliable passive trinocular stereovision," in *Proc. First Int. Conf. Comput. Vision* (London), June 1987, pp. 422–427.

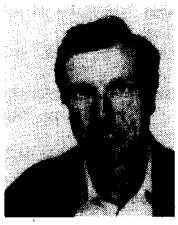
[9] O. Faugeras *et al.*, "Toward a flexible vision system," in *Robot Vision* (A. Pugh, Ed.). London: IFS, 1983, pp. 129–142, ch. 3.
[10] B. Horn and B. Schunk, "Determining optical flow," *Artif. Intell.*, vol. 20, pp. 199–228, 1981.
[11] H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Comput. Vision Graphics Image Processing*, vol. 21, pp. 85–117, 1983.
[12] E. Hildreth, *The Measurement of Visual Motion*. Cambridge, MA: MIT Press, 1983.
[13] A. Bruss and B. Horn, "Passive navigation," *Comput. Vision Graphics Image Processing*, vol. 21, pp. 3–20, Jan. 1983.
[14] G. Adiv, "Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.* (San Francisco, CA), June 1985, pp. 70–77.
[15] S. Maybank, "A theoretical study of optical flow," Ph.D. thesis, Birkbeck College, Univ. of London, 1987.
[16] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: MIT Press, 1979.
[17] R. Tsai and T. Huang, "Estimating 3-D motion parameters of a rigid planar patch, i," *IEEE Trans. Acous. Speech Signal Processing*, vol. ASSP-29, pp. 1147–1152, Dec. 1981.
[18] T. Huang and R. Tsai, "Image sequence analysis: Motion estimation," in *Image Sequence Analysis*, (T. Huang, Ed.). Berlin, Heidelberg: Springer, 1981, pp. 1–18, ch. 3.
[19] H. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133–135, 1981.
[20] A. Mitiche, S. Seida, and J. Aggarwal, "Line based computation of structure and motion using angular invariance," in *Proc. Workshop Motion: Representation Anal.* (Charleston, SC), May 1986, pp. 175–180.
[21] O. Faugeras, F. Lustman, and G. Toscani, "Motion and structure from motion from point and line matches," in *Proc. First Int. Conf. Comput. Vision* (London), 1987, pp. 25–34.
[22] O. Faugeras and S. Maybank, "Motion from point matches: multiplicity of solutions," *Int. J. Comput. Vision*, vol. 4, pp. 225–246, June 1990.
[23] T. Broida and R. Chellappa, "Experiments and uniqueness results on object structure and kinematics from a sequences of monocular images," in *Proc. IEEE Workshop Visual Motion* (Irvine, CA), Mar. 1989, pp. 21–30.
[24] J. Weng, T. Huang, and N. Ahuja, "3-D motion estimation, understanding, and prediction from noisy image sequences," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 3, pp. 370–389, 1987.
[25] G. Young and R. Chellappa, "3-D motion estimation using a sequence of noisy stereo images: Models, estimation, and uniqueness results," *IEEE Trans. Patt. Anal. Macvhine Intell.*, vol. 12, pp. 735–759, Aug. 1990.
[26] Z. Zhang and O. Faugeras, "Tracking and grouping 3-D line segments," in *Proc. Third Int. Conf. Comput. Vision* (Osaka, Japan), Dec. 1990, pp. 577–580.
[27] H. Shariat and K. Price, "Motion estimation with more than two frames," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, pp. 417–434, May 1990.
[28] T. Broida, S. Chandrashekhar, and R. Chellappa, "Recursive 3-D motion estimation from a monocular image sequence," *IEEE Trans. Aero. Electron. Syst.*, vol. 26, pp. 639–656, July 1990.
[29] S. Pollard, J. Porrill, J. Mayhew, and J. Frisby, "Matching geometrical descriptions in three-space," *Image Vision Comput.*, vol. 5, pp. 73–78, May 1987.
[30] R. Bolles and R. Cain, "Recognizing and locating partially visible objects: The local-feature-focus method," *Int. J. Robotics Res.*, vol. 1, no. 3, pp. 57–82, 1982.
[31] P. Horaud and R. Bolles, "3DPO's strategy for matching three-dimensional objects in range data," in *Proc. Int. Conf. Robotics Automat.* (Atlanta, GA), Mar. 13–15, 1984, pp. 78–85.
[32] W. Grimson and T. Lozano-Perez, "Model-based recognition and localization from sparse range or tactile data," *Int. J. Robotics Res.*, vol. 5, pp. 3–34, Fall 1984.
[33] H. Chen and T. Huang, "Maximal matching of 3-D points for multiple-object motion estimation," *Patt. Recog.*, vol. 21, no. 2, pp. 75–90, 1988.
[34] ——, "An algorithm for matching 3-D line segments with application to multiple–object motion estimation," in *Proc. IEEE Workshop Comput. Vision*, Nov. 30–Dec. 2, 1987, pp. 151–156.
[35] Y. Kim and J. Aggarwal, "Determining object motion in a sequence of stereo images," *IEEE J. Robotics Automat.*, vol. RA-3, pp. 599–614, Dec. 1987.
[36] S. Blostein and T. Huang, "Estimation 3-D motion from range data," in *Proc. First Conf. Artif. Intell. Applications* (Denver, CO), Dec. 1984, pp. 246–250.
[37] T. Huang and S. Blostein, "Robust algorithms for motion estimation based on two sequential stereo image pairs," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.* (San Francisco, CA), 1985, pp. 518–523.
[38] O. Faugeras and M. Hebert, "The representation, recognition, and locating of 3-D shapes from range data," *Int. J. Robotics Res.*, vol. 5, no. 3, pp. 27–52, 1986.
[39] O. Faugeras, N. Ayache, and B. Faverjon, "Building visual maps by combining noisy stereo measurements," in *Proc. Int. Conf. Robotics Automat.* (San Francisco, CA), Apr. 1986, pp. 1433–1438.
[40] N. Ayache and O. Faugeras, "Building, registrating and fusing noisy visual maps," in *Proc. First Int. Conf. Comput. Vision* (London), June 1987, pp. 73–82.
[41] Z. Lin, H. Lee, and T. Huang, "Finding 3-D point correspondence in motion estimation," in *Proc. 8th Int. Conf. Patt. Recog.* (Paris, France), 1986, pp. 303–305.
[42] W. Grimson and T. Lozano-Perez, "Localizing overlapping parts by searching the interpretation tree," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-7, no. 4, pp. 469–482, 1987.
[43] N. Ayache and O. Faugeras, "HYPER: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, pp. 44–54, Jan. 1986.
[44] D. Lowe, *Perceptual Organization and Visual Recognition*. Boston, MA: Kluwer, 1985.
[45] ——, "The viewpoint consistency constraint," *Int. J. Comput. Vision*, no. 1, pp. 57–72, 1987.
[46] O. Faugeras, N. Ayache, and Z. Zhang, "A preliminary investigation of the problem of determining ego- and object motions from stereo," in *Proc. 9th Int. Conf. Patt. Recog.* (Rome, Italy), 1988, pp. 242–246.
[47] Z. Zhang, O. Faugeras, and N. Ayache, "Analysis of a sequence of stereo scenes containing multiple moving objects using rigidity constraints," in *Proc. Second Int. Conf. Comput. Vision* (Tampa, FL), Dec. 1988, pp. 177–186.
[48] Z. Zhang, "Motion analysis from a sequence of stereo frames and its applications," Ph.D. thesis, Univ of Paris, Orsay, Paris, France, 1990, in English.
[49] H. Durrant-Whyte, "Uncertain geometry in robotics," *IEEE J. Robotics Automat.*, vol. 4, pp. 23–31, Feb. 1988.
[50] N. Ayache and O. Faugeras, "Maintaining representations of the environment of a mobile robot," in *Proc. Int. Symp. Robotics Res.* (Santa Cruz, CA), Aug. 1987.
[51] N. Ayache, *Artificial Vision for Mobile Robots: Stereo Vision and Multisensory Perception*. Cambridge, MA: MIT Press, 1991.
[52] N. Ayache and O. D. Faugeras, "Maintaining representations of the environment of a mobile robot," *IEEE Trans. Robotics Automat.*, vol. 5, pp. 804–819, Dec. 1989.
[53] K. Roberts, "A new representation for a line," in *Proc. IEEE Conf. Comput. Vision Patt. Recog.* (Ann Arbor, MI), June 1988, pp. 635–640.
[54] Z. Zhang and O. Faugeras, "Building a 3-D world model with a mobile robot: 3-D line segment representation and integration," in *Proc. 10th Int. Conf. Patt. Recog.* (Atlantic City, NJ), June 1990, pp. 38–42.
[55] D. Murray and D. Cook, "Using the orientation of fragmentary 3-D edge segments for polyhedral object recognition," *Int. J. Comput. Vision*, no. 2, pp. 153–169, 1988.
[56] K. Arun, T. Huang, and S. Blostein, "Least-squares fitting of two 3-D point sets," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, pp. 698–700, Sept. 1987.
[57] Z. Zhang and O. Faugeras, "Determining motion from 3-D line segments: A comparative study," in *Proc. Brit. Machine Vision Conf.* (London), Sept. 24–27, 1990, pp. 85–90.
[58] A. Jazwinsky, *Stochastic Processes and Filtering Theory*. New York: Academic, 1970.
[59] P. Maybeck, *Stochastic Models, Estimation and Control*. New York: Academic, 1979, vol. 1.
[60] ——, *Stochastic Models, Estimation and Control*. New York: Academic, 1982, vol. 2.
[61] B. Horn, *Robot Vision*. New York: McGraw–Hill, 1986.
[62] O. D. Faugeras, "A few steps toward artificial 3-D vision," in *Robotics Science*, (M. Brady, Ed.). Cambridge, MA: MIT Press, 1989, pp. 39–137, ch. 2.
[63] O. Rodrigues, "Des lois géométriques qui régissent les déplacements d'un systéme solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire," *J. Mathématiques Pures et Appliquées*, vol. 5, pp. 380–440, 1840.
[64] Z. Zhang and O. Faugeras, "Calibration of a mobile robot with application to visual navigation," in *Proc. IEEE Workshop Visual Motion* (Irvine, CA), Mar. 1989, pp. 306–313.



Zhengyou Zhang was born in Zhejiang, China, on April 1, 1965. He received the B. S. degree in electronic engineering from the University of Zhejiang, China, in 1985, the D. E. A. diploma in computer science from the University of Nancy, France, in 1987, and the Ph.D. degree in computer science from the University of Paris XI, Orsay, France, in 1990.

From 1987 to 1990, he was a research assistant in the Computer Vision and Robotics Group of the Insitut National de Recherche en Informatique et en Automatique (INRIA), France. Currently, he is a research scientist at INRIA. His current research interests include computer vision, mobile robotics, dynamic scene analysis, and multisensor fusion.

Dr. Zhang is the coauthor (with O. Faugeras) of the book *3-D Dynamic Scene Analysis: A Stereo Based Approach* (Springer: Berlin, Heidelberg, 1992).



Olivier Faugeras (SM'90) is Research Director at the National Research Institute in Computer Science and Control Theory (INRIA), Valbonne, France, where he leads the Computer Vision and Robotics Group. His research interests include the application of mathematics to computer vision, robotics, shape representation, computational geometry, and the architectures for artificial vision systems as well as the links between artificial and biological vision. He is also an Associate Professor of Applied Mathematics at the Ecole Polytechnique in Palaiseau, France, where he teaches computer science, computer vision, and computational geometry.

Dr. Faugeras is an associate editor for several international scientific journals including *The International Journal of Computer Vision, The International Journal of Robotics Research, Pattern Recognition Letters, Signal Processing, Robotics and Autonomous Systems*, and *Vision Research*. From 1987–1990, he served as Associate Editor for IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE. In April 1989, he received the "Institut de France-Fondation Fiat" prize from the French Science Academy for his work in vision and robotics.