Determining Motion from 3D Line Segment Matches: A Comparative Study

Zhengyou Zhang **Olivier D. Faugeras**

INRIA Sophia-Antipolis, 2004 route des Lucioles
06565 Valbonne Cedex - France

Motion estimation is a very important problem in dynamic scene analysis. Although it is easier to estimate motion parameters from 3D data than from 2D images, it is not trivial since the 3D data we have are almost always corrupted by noise. This article presents a comparative study on motion estimation from 3D line segments. Two representations of line segments and two representations of rotation are described. With different representations of line segments and rotation, a number of methods for motion estimation are presented, including the Extended Kalman Filter, a general Minimization process and the Singular Value Decomposition. These methods are compared using both synthetic and real data obtained by a trinocular stereo. We observe that the Extended Kalman Filter with the rotation axis representation of rotation is preferable. We note that all methods discussed in this article can be directly applied to 3D point data.

Keywords: *Motion Estimation, Motion from Stereo, Noisy System, Nonlinear System, Minimization.*

Although it is easier to estimate motion parameters from 3D data than from 2D (monocular) images, it is not trivial since the 3D data we have are almost always corrupted by noise. A number of methods are proposed to combat the noise. To our knowledge, no work has been carried out to compare those methods. We believe that this work is important for researchers working on motion analysis to choose an appropriate approach based on the efficiency, accuracy and robustness. In this paper, we present a comparative study of different methods of determining motion from correspondences of 3D line segments. The reader is referred to [13, 14] for methods to recover 3D line segment matches. For the problem of determining motion from 3D point correspondences, the reader is referred to [5, 10, 8, 3]. [8] presents also a method to determine motion from planar patches.

The paper is organized as follows. First, two representations for line segments are presented and the problem we address is formulated. Second, representations of motion are described. Third, we show how to estimate motion using the Extended Kalman Filter, minimization techniques and an analytical solution. Finally we provide the results of our comparative study on different methods.

PROBLEM STATEMENT

In this section, we give two representations of line segments and the transformation of line segments under rigid motion. Finally, we formalize the problem we should solve.

Line segment representations

3D line segments can be reconstructed from stereo [4] or extracted from range data. A common approach is to represent a 3D line segment L by its endpoints, noted as M_1 and M_2. Equivalently, L can be represented by two vectors $(\mathbf{l}, \mathbf{m})$, $\mathbf{l}$ is the non-normalized direction vector of L and $\mathbf{m}$, the midpoint of L, that is:

$$\mathbf{l} = M_2 - M_1, \quad \mathbf{m} = (M_1 + M_2)/2. \tag{1}$$

Besides of the geometric parameters, the uncertainty is also manipulated in our stereo system. Suppose that W_1 and W_2 are the covariance matrices of M_1 and M_2, respectively, and suppose that M_1 and M_2 are independent, we can compute the covariance matrix of $(\mathbf{l}, \mathbf{m})$ as follows:

$$W_L = \begin{bmatrix} W_1 + W_2 & \frac{W_2 - W_1}{2} \\ \frac{W_2 - W_1}{2} & \frac{W_1 + W_2}{4} \end{bmatrix}, \tag{2}$$

which is a 6 × 6 matrix.

We can also represent a 3D line segment by its infinite supporting line. There are a number of ways to represent a 3D line, one of which is to represent the line by two vectors $(\mathbf{u}, \mathbf{d})$, where $\mathbf{u}$ is its unit direction vector, the norm of $\mathbf{d}$ gives the distance from the origin to the line and the direction of $\mathbf{d}$ is parallel to the normal of the plane passing the line and the origin (see Figure 1). We can compute $(\mathbf{u}, \mathbf{d})$ from the above representation $(\mathbf{l}, \mathbf{m})$. Indeed, we have:

$$\mathbf{u} = \mathbf{l}/\|\mathbf{l}\| \quad \text{and} \quad \mathbf{d} = \mathbf{u} \wedge \mathbf{m} = (\mathbf{l} \wedge \mathbf{m})/\|\mathbf{l}\|, \tag{3}$$

where $\wedge$ is the cross product. In reality, $\mathbf{m}$ may be any point on the line. As we see later, we use this representation to derive an analytical solution of motion.

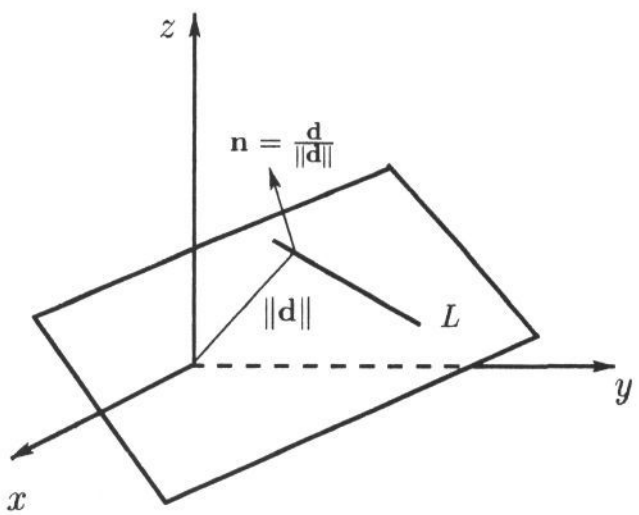

Fig. 1: A 3D line representation

We should point out that segments addressed in this paper are *oriented*, which is obtained in stereo through the

BMVC 1990 doi:10.5244/C.4.17

information about the intensity contrast.

3D line segment transformation

If a 3D line segment undergoes a rigid displacement $(\mathbf{R}, \mathbf{t})$ and if we use the first representation(see Equation 1), let $(\mathbf{l}, \mathbf{m})$ be the parameters before transformation and $(\mathbf{l}', \mathbf{m}')$, that after transformation, following relations hold:

$$\mathbf{l}' = \mathbf{Rl} \quad \text{and} \quad \mathbf{m}' = \mathbf{Rm} + \mathbf{t}. \tag{4}$$

As we know that a segment can be differently segmented in successive views and that the direction of segment is relatively more stable, we insist on having the transformed segment parallel to the segment in the second view which yields [1]:

$$\mathbf{l}' \wedge \mathbf{Rl} = 0 \quad \text{and} \quad \mathbf{l}' \wedge (\mathbf{m}' - \mathbf{Rm} - \mathbf{t}) = 0. \tag{5}$$

We shall use these two equations to compute the motion parameters. If we note:

$$f(\mathbf{p}, \mathbf{x}) = \begin{bmatrix} \mathbf{l}' \wedge \mathbf{Rl} \\ \mathbf{l}' \wedge (\mathbf{m}' - \mathbf{Rm} - \mathbf{t}) \end{bmatrix}, \tag{6}$$

Equation 5 becomes:

$$f(\mathbf{p}, \mathbf{x}) = 0, \tag{7}$$

where $\mathbf{x} = [\mathbf{l}^t, \mathbf{m}^t, \mathbf{l}'^t, \mathbf{m}'^t]^t$ (the superscript ^{t} denotes the transpose of a vector or a matrix) is a 12 dimensional vector which we call *measurement vector* and $\mathbf{p}$ is one of the motion parametrizations described below. In filtering terminology, $\mathbf{p}$ is called *state vector*. So our problem can be formulated as follows:

Given *n measurement vectors:* $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n$,
i.e., given a noisy system:
$f(\mathbf{p}, \mathbf{x}_i) = 0$, *for* $i = 1, ..., n$,
Recover *the motion parameters* $\mathbf{p}$.

MOTION REPRESENTATION

Any rigid motion can be uniquely decomposed into a rotation around an axis passing through the origin of the coordinate system, and a translation. The translation is supposed after the rotation. A translation can be simply represented by a 3 dimensional vector $\mathbf{t} = [t_x\ t_y\ t_z]^t$. A rotation can be represented by a 3×3 matrix $\mathbf{R}$ called *rotation matrix*, which is an orthogonal matrix satisfying:

$$\mathbf{R}\mathbf{R}^t = \mathrm{I}. \tag{8}$$

This representation gives a simple way of representing a 3D rotation but leads to a high dimensional space of constraints. Several other representations of rotation are available [7] and we present here two of them: using the rotation axis and using a quaternion.

Using the rotation axis

A rotation can be defined as a three dimensional vector $\mathbf{r} = [a\ b\ c]^t$ whose direction is that of the rotation axis and whose norm is equal to the rotation angle.

For convenience, we note $\widetilde{\mathbf{v}}$ as the antisymmetric matrix defined by $\mathbf{v}$. More precisely, if $\mathbf{v} = [x\ y\ z]^t$, then

$$\widetilde{\mathbf{v}} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix}. \tag{9}$$

In fact, for any three dimensional vectors $\mathbf{u}$ and $\mathbf{v}$, we have $\mathbf{u} \wedge \mathbf{v} = \widetilde{\mathbf{u}}\mathbf{v}$.

The relation between $\mathbf{R}$ and $\mathbf{r}$ is the following Rodrigues formula:

$$\mathbf{R} = e^{\widetilde{\mathbf{r}}} = \mathrm{I}_3 + \mathrm{f}(\theta)\widetilde{\mathbf{r}} + \mathrm{g}(\theta)\widetilde{\mathbf{r}}^2, \tag{10}$$

where $\theta = \sqrt{a^2 + b^2 + c^2}$ is the rotation angle, $\mathrm{f}(\theta) = \frac{\sin\theta}{\theta}$ and $\mathrm{g}(\theta) = \frac{1-\cos\theta}{\theta^2}$.

So, the parameter vector $\mathbf{p}$ in Equation 7 is in this representation a 6 dimensional vector, noted as $\mathbf{s}$:

$$\mathbf{s} = \begin{bmatrix} \mathbf{r} \\ \mathbf{t} \end{bmatrix}. \tag{11}$$

Using a quaternion

Quaternions have been found useful in Robotics and Vision [9]. A quaternion $\mathbf{q}$ can be considered as being either a 4 dimensional vector $[\lambda_0\ \lambda_1\ \lambda_2\ \lambda_3]^t$ or as a pair $(\alpha, \boldsymbol{\gamma})$ where α is a real number equal to λ_0, and $\boldsymbol{\gamma}$ is the vector $[\lambda_1\ \lambda_2\ \lambda_3]^t$. We define the multiplication $\times$ of two quaternions $\mathbf{q}$ and $\mathbf{q}'$ as follows:

$$\mathbf{q} \times \mathbf{q}' = (\alpha\alpha' - \boldsymbol{\gamma} \cdot \boldsymbol{\gamma}', \alpha\boldsymbol{\gamma}' + \alpha'\boldsymbol{\gamma} + \boldsymbol{\gamma} \wedge \boldsymbol{\gamma}'). \tag{12}$$

The conjugate and the magnitude of a quaternion $\mathbf{q}$ are defined as follows:

$$\overline{\mathbf{q}} = (\alpha, -\boldsymbol{\gamma}), \tag{13}$$

$$|\mathbf{q}|^2 = \mathbf{q} \times \overline{\mathbf{q}} = (\alpha^2 + \|\boldsymbol{\gamma}\|^2, \mathbf{0}) = (\|\mathbf{q}\|^2, \mathbf{0}). \tag{14}$$

A real number x is identified with the quaternion $(x, \mathbf{0})$ and a 3 dimensional vector $\mathbf{v}$ is identified with the quaternion $(0, \mathbf{v})$.

A rotation can then be represented by two quaternions $\mathbf{q} = (\alpha, \boldsymbol{\gamma})$ and $-\mathbf{q}$, with $|\mathbf{q}| = 1$. The relation between this representation and the rotation axis one is:

$$\alpha = \cos(\theta/2) \quad \text{and} \quad \boldsymbol{\gamma} = \sin(\theta/2)\mathbf{u}, \tag{15}$$

where $\theta = \|\mathbf{r}\|$ and $\mathbf{u} = \mathbf{r}/\|\mathbf{r}\|$. Note that there are two quaternions for one rotation. It is not surprising since a rotation of angle θ around an axis $\mathbf{u}$ is the same as a rotation of angle $2\pi - \theta$ around the axis $-\mathbf{u}$. Usually, the rotation angle between two successive views does not beyond π, so we can impose that the first element α of quaternion $\mathbf{q}$ must be positive. Thus the mapping between rotation and quaternion is unique under this new constraint.

The relation between $\mathbf{R}$ and an unit quaternion $\mathbf{q} = [\lambda_0, \lambda_1, \lambda_2, \lambda_3]^t$ is given as follows:

$$\mathbf{R} = \begin{bmatrix} \lambda_0^2 + \lambda_1^2 - \lambda_2^2 - \lambda_3^2 & 2(\lambda_1\lambda_2 - \lambda_0\lambda_3) & 2(\lambda_1\lambda_3 + \lambda_0\lambda_2) \\ 2(\lambda_1\lambda_2 + \lambda_0\lambda_3) & \lambda_0^2 - \lambda_1^2 + \lambda_2^2 - \lambda_3^2 & 2(\lambda_2\lambda_3 - \lambda_0\lambda_1) \\ 2(\lambda_1\lambda_3 - \lambda_0\lambda_2) & 2(\lambda_2\lambda_3 + \lambda_0\lambda_1) & \lambda_0^2 - \lambda_1^2 - \lambda_2^2 + \lambda_3^2 \end{bmatrix}$$

The product $\mathbf{Rv}$ can be identified as the product of quaternions:

$$\mathbf{Rv} = \mathbf{q} \times \mathbf{v} \times \overline{\mathbf{q}}. \tag{16}$$

So, the parameter vector $\mathbf{p}$ in Equation 7 is in this representation a 7 dimensional vector, noted as $\mathbf{s}_q$:

$$\mathbf{s}_q = \begin{bmatrix} \mathbf{q} \\ \mathbf{t} \end{bmatrix}, \tag{17}$$

under the constraint $\|\mathbf{q}\| = 1$.

Derivation of rotation matrix

At this point, we have two parametrizations for a rotation. In this section, we give the first derivatives of the rotation matrix with respect to each of its parametrizations, which are used in the methods described below. Specially, let $\mathbf{v} = [v_0\ v_1\ v_2]^t$ be a 3 dimensional vector, we are interested in computing the derivative of $\mathbf{Rv}$ about $\mathbf{r}$ and that about $\mathbf{q}$.

The derivative of $\mathbf{Rv}$ about $\mathbf{r}$ is:

$$\frac{\partial \mathbf{Rv}}{\partial \mathbf{r}} = \frac{\cos(\theta) - \mathrm{f}(\theta)}{\theta^2}(\widetilde{\mathbf{rv}})\mathbf{r}^t + \frac{\sin(\theta) - 2\theta \mathrm{g}(\theta)}{\theta^3}(\widetilde{\mathbf{r}}(\widetilde{\mathbf{rv}}))\mathbf{r}^t - \mathrm{f}(\theta)\widetilde{\mathbf{v}} + \mathrm{g}(\theta)(-(\widetilde{\mathbf{r} \wedge \mathbf{v}}) + (\mathbf{r}.\mathbf{v})\mathbf{I}_3 - \mathbf{v}\mathbf{r}^t). \quad (18)$$

where $\mathrm{f}(\theta)$ and $\mathrm{g}(\theta)$ have the same definition as above. We denote $E(\mathbf{R}, \mathbf{v})$, a 3×3 matrix for $\frac{\partial \mathbf{Rv}}{\partial \mathbf{r}}$.

The derivative of $\mathbf{Rv}$ about $\mathbf{q}$ is simpler. Indeed, we have:

$$\frac{\partial \mathbf{Rv}}{\partial \mathbf{q}} = \begin{bmatrix} d_0 & d_1 & d_2 & d_3 \\ -d_3 & -d_2 & d_1 & d_0 \\ d_2 & -d_3 & -d_0 & d_1 \end{bmatrix}, \quad (19)$$

where

$$\begin{aligned} d_0 &= 2(\lambda_0 v_0 - \lambda_3 v_1 + \lambda_2 v_2), \\ d_1 &= 2(\lambda_1 v_0 + \lambda_2 v_1 + \lambda_3 v_2), \\ d_2 &= 2(-\lambda_2 v_0 + \lambda_1 v_1 + \lambda_0 v_2), \\ d_3 &= 2(-\lambda_3 v_0 - \lambda_0 v_1 + \lambda_1 v_2). \end{aligned} \quad (20)$$

We denote $Q(\mathbf{R}, \mathbf{v})$, a 3×4 matrix for $\frac{\partial \mathbf{Rv}}{\partial \mathbf{q}}$.

The derivative of $f(\mathbf{p}, \mathbf{x})$ (See Equation 7) with respect to $\mathbf{s}$ can be easily computed as follows:

$$\frac{\partial f(\mathbf{p}, \mathbf{x})}{\partial \mathbf{s}} = \begin{bmatrix} \widetilde{\mathbf{l}'} E(\mathbf{R}, \mathbf{l}) & 0 \\ -\widetilde{\mathbf{l}'} E(\mathbf{R}, \mathbf{m}) & -\widetilde{\mathbf{l}'} \end{bmatrix}, \quad (21)$$

and the derivative of $f(\mathbf{p}, \mathbf{x})$ about $\mathbf{s}_q$ is:

$$\frac{\partial f(\mathbf{p}, \mathbf{x})}{\partial \mathbf{s_q}} = \begin{bmatrix} \widetilde{\mathbf{l}'} Q(\mathbf{R}, \mathbf{l}) & 0 \\ -\widetilde{\mathbf{l}'} Q(\mathbf{R}, \mathbf{m}) & -\widetilde{\mathbf{l}'} \end{bmatrix}. \quad (22)$$

The tilde "~" above is defined as in Equation 9.

ESTIMATING MOTION USING EXTENDED KALMAN FILTER

Kalman Filter is a powerful tool to deal with a linear noisy system. For details of this filter, the reader is referred to elsewhere [11].

But Kalman Filter is not directly applicable to our problem, because Equation 7 is non-linear. So we use the so-called Extended Kalman Filter [6, 2], that is, we linearize first Equation 7, then we apply the Kalman Filter on the linearized system. The linearized system is as follows:

$$\mathbf{p}_i = \mathbf{p}_{i-1}, \quad (23)$$

$$\mathbf{y}_i = M_i \mathbf{p}_i + \mathbf{w}_i, \quad (24)$$

where

$$\mathbf{y}_i = -f_i(\hat{\mathbf{p}}_{i-1}, \hat{\mathbf{x}}_i) + \frac{\partial f_i}{\partial \mathbf{p}} \hat{\mathbf{p}}_{i-1},$$

$$M_i = \frac{\partial f_i}{\partial \mathbf{p}},$$

$$\mathbf{w}_i = \frac{\partial f_i}{\partial \mathbf{x}}(\mathbf{x}_i - \hat{\mathbf{x}}_i).$$

$\hat{\mathbf{p}}_{i-1}$ is the current estimation of $\mathbf{p}$ before processing the linearized system, and $\hat{\mathbf{x}}_i$ is the current measurement. The derivative of $f(\mathbf{p}, \mathbf{x})$ about $\mathbf{p}$ can be computed using either Equation 21 or Equation 22. What we need to compute yet is the derivative of $f(\mathbf{p}, \mathbf{x})$ with respect to $\mathbf{x}$, which is the same for both motion parametrizations:

$$\frac{\partial f(\mathbf{p}, \mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} \widetilde{\mathbf{l}'}\mathbf{R} & 0 & -(\widetilde{\mathbf{Rl}}) & 0 \\ 0 & -\widetilde{\mathbf{l}'}\mathbf{R} & \widetilde{\mathbf{Rm}} + \widetilde{\mathbf{t}} - \widetilde{\mathbf{m}'} & \widetilde{\mathbf{l}'} \end{bmatrix}. \quad (25)$$

So, the expectation and the covariance of the new measurement noise $\mathbf{w}_i$ are easily derived from that of $\mathbf{x}_i$ as:

$$E[\mathbf{w}_i] = 0, \quad \text{and} \quad W_i \stackrel{\text{def}}{=} E[\mathbf{w_i}\mathbf{w_i}^t] = \frac{\partial f_i}{\partial \mathbf{x}_i} \Lambda_i \frac{\partial f_i}{\partial \mathbf{x}_i}^t,$$

where Λ_i is the covariance matrix of $\mathbf{x}_i$. If we suppose that L_i and L_i' are independent, then:

$$\Lambda_i = \begin{bmatrix} W_{L_i} & 0 \\ 0 & W_{L_i'} \end{bmatrix}.$$

An important remark is that when we use the quaternion to represent the rotation, we have added the constraint $\|\mathbf{q}\| = 1$, *i.e.*, $\mathbf{q}^t\mathbf{q} = 1$ in Equation 7 as an additional measurement.

ESTIMATING MOTION USING MINIMIZATION

We can restate the motion estimation problem as a minimization problem, that is:

Given *n measurement vectors:* $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$,
Recover *the motion parameters* $\mathbf{p}$ *so that*

$$\sum_{i=1}^{n} [f(\mathbf{p}, \mathbf{x}_i)^t f(\mathbf{p}, \mathbf{x}_i)]$$

is minimized.

Here $\sum_{i=1}^{n} [f(\mathbf{p}, \mathbf{x}_i)^t f(\mathbf{p}, \mathbf{x}_i)]$ is called *objective function* in the minimization problem which is denoted by $\mathcal{F}(\mathbf{p}, \mathbf{x}_i)$, *i.e.*,

$$\mathcal{F}(\mathbf{p}, \mathbf{x}_i) = \sum_{i=1}^{n} [(\mathbf{l}_i' \wedge \mathbf{Rl}_i)^t (\mathbf{l}_i' \wedge \mathbf{Rl}_i) + (\mathbf{l}_i' \wedge (\mathbf{m}_i' - \mathbf{Rm}_i - \mathbf{t}))^t (\mathbf{l}_i' \wedge (\mathbf{m}_i' - \mathbf{Rm}_i - \mathbf{t}))]. \quad (26)$$

There exist many routines in mathematical libraries (like **Nag** [12]) to minimize $\mathcal{F}(\mathbf{p}, \mathbf{x}_i)$ with or without constraints. The minimization process can be speeded up if we supply also the first derivations, which can be easily obtained using Equation 21 or 22.

We can adopt the weighted least-squares method to take into account the uncertainty of measurements. That is, we first use the general minimization algorithms to obtain a better estimate of motion by minimizing the simple criterion (Equation 26). Then we compute the covariance of $f_i(\mathbf{p}, \mathbf{x_i})$ using the first order approximation, denoted by W_i. We again use the minimization algorithm to obtain a new estimate by minimizing:

$$\sum_{i=1}^{n} [f(\mathbf{p}, \mathbf{x}_i)^t W_i^{-1} f(\mathbf{p}, \mathbf{x}_i)]. \quad (27)$$

The new estimate is generally better than the previous one, but more time is needed for the compensation.

ANALYTICAL SOLUTION: SVD

The second representation of line segments (See Equation 3) and the quaternion representation of rotation are used. [8] has already proposed an analytical method from point and plan correspondences. The method described here is directly inspired from theirs.

The relation between a line segment $(\mathbf{u}, \mathbf{d})$ and the transformed line segment $(\mathbf{u}', \mathbf{d}')$ is:

$$\mathbf{u}' = \mathbf{Ru}, \quad (28)$$

$$\mathbf{d}' = \mathbf{Rd} + \mathbf{u}' \wedge \mathbf{t}. \quad (29)$$

The first one is evident, while the second one can be easily verified using the definition of $\mathbf{d}$:

$$\begin{aligned} \mathbf{d}' &\equiv \mathbf{u}' \wedge \mathbf{m}' = \mathbf{u}' \wedge (\mathbf{Rm} + \mathbf{t}) = \mathbf{Ru} \wedge \mathbf{Rm} + \mathbf{u}' \wedge \mathbf{t} \\ &= \mathbf{R}(\mathbf{u} \wedge \mathbf{m}) + \mathbf{u}' \wedge \mathbf{t} = \mathbf{Rd} + \mathbf{u}' \wedge \mathbf{t}. \end{aligned} \quad (30)$$

Due to the fact that the orientation of a segment is more conservative than other parameters (for example, $\mathbf{d}$ or $\mathbf{m}$),

we divide the motion determination problem into two subproblems:

1. Determine first the rotation using Equation 28 under the following criterion:
$$\mathbf{Min} \sum_{i=1}^{n} \| \mathbf{u}'_i - \mathbf{R}\mathbf{u}_i \|^2. \quad (31)$$
2. Determine then the translation using Equation 29 under the following criterion:
$$\mathbf{Min} \sum_{i=1}^{n} \| \mathbf{d}'_i - \mathbf{R}^* \mathbf{d}_i - \mathbf{u}'_i \wedge \mathbf{t} \|^2, \quad (32)$$
where $\mathbf{R}^*$ is the rotation matrix recovered in the first step.

Determining rotation

By using Equation 16, we can restate the minimization problem of Equation 31 in quaternion notation as:
$$\mathbf{Min} \sum_{i=1}^{n} \| \mathbf{u}'_i - \mathbf{q} \times \mathbf{u}_i \times \overline{\mathbf{q}} \|^2 \quad (33)$$
subject to the constraint $\|\mathbf{q}\| = 1$. Since the module is multiplicative and $\|\mathbf{q}\| = 1$, Equation 33 can be rewritten as:
$$\mathbf{Min} \sum_{i=1}^{n} \| \mathbf{u}'_i \times \mathbf{q} - \mathbf{q} \times \mathbf{u}_i \|^2. \quad (34)$$
From the definition of product of two quaternions, we can express $\mathbf{u}'_i \times \mathbf{q} - \mathbf{q} \times \mathbf{u}_i$ as a linear function of $\mathbf{q}$. Indeed, there exists a matrix A_i such that:
$$\mathbf{u}'_i \times \mathbf{q} - \mathbf{q} \times \mathbf{u}_i = A_i \mathbf{q}, \quad (35)$$
where
$$A_i = \begin{bmatrix} 0 & (\mathbf{u}_i - \mathbf{u}'_i)^t \\ -(\mathbf{u}_i - \mathbf{u}'_i) & (\mathbf{u}_i + \mathbf{u}'_i) \end{bmatrix}.$$
Then, Equation 33 can be further restated as:
$$\mathbf{Min} \sum_{i=1}^{n} \mathbf{q}^t A_i^t A_i \mathbf{q} = \mathbf{Min}\ \mathbf{q}^t A \mathbf{q}, \quad (36)$$
where $A = \sum_{i=1}^{n} A_i^t A_i$ and $\|\mathbf{q}\| = 1$. The matrix A can be computed incrementally.

Since A is a symmetric matrix, the solution to this problem is the four dimensional vector $\mathbf{q}_{\min}$ corresponding to the smallest eigenvalue of A.

Determining translation

We can determine translation using the standard minimization technique. Let the derivation of Equation 32 about $\mathbf{t}$ be zero, we have:
$$\sum_{i=1}^{n} 2(\mathbf{d}'_i - \mathbf{R}^* \mathbf{d}_i - \widetilde{\mathbf{u}'_i}\mathbf{t})^t \widetilde{\mathbf{u}'_i} = 0, \quad (37)$$
that is,
$$\left(\sum_{i=1}^{n} \widetilde{\mathbf{u}'_i} (\widetilde{\mathbf{u}'_i})^t \right) \mathbf{t} = \sum_{i=1}^{n} \widetilde{\mathbf{u}'_i}^t (\mathbf{d}'_i - \mathbf{R}^* \mathbf{d}_i). \quad (38)$$
If $\left(\sum_{i=1}^{n} \widetilde{\mathbf{u}'_i} (\widetilde{\mathbf{u}'_i})^t \right)$ is a full rank matrix, we have explicitly the translation vector:
$$\mathbf{t} = \left(\sum_{i=1}^{n} \widetilde{\mathbf{u}'_i} (\widetilde{\mathbf{u}'_i})^t \right)^{-1} \left(\sum_{i=1}^{n} \widetilde{\mathbf{u}'_i}^t (\mathbf{d}'_i - \mathbf{R}^* \mathbf{d}_i) \right). \quad (39)$$
If not, $\mathbf{t}$ is unrecoverable. It can be shown that $\left(\sum_{i=1}^{n} \widetilde{\mathbf{u}'_i} (\widetilde{\mathbf{u}'_i})^t \right)$ is always of full rank if two of $\mathbf{u}'_i$ ($i = 1..n$) are different (non-parallel).

EXPERIMENTAL RESULTS

The objective of our comparative study is to investigate the applicability of different methods on stereo data. Due to space limit, only *part* of the results using synthetic data are provided. See [14] for more results and the comparisons using real stereo data. For simplicity, we use some abbreviations to refer different methods (cf. Table 1). Note that we have not implemented the weighted least-squares method in using the general minimization techniques.

Abbrev.	*Rep. of rot.*	*Approach*
EKF-AXIS	rotation axis	extended Kalman filter
EKF-QUAT	quaternion	extended Kalman filter
MIN-AXIS	rotation axis	Gauss-Newton minimiz.
MIN-QUAT	quaternion	seq. quad. programming
EIGEN	quaternion	singular value decomp.

Table 1: Abbreviations of different methods

The synthetic data we used contain 26 segments. One of the endpoints of each segment is fixed at the center of a sphere with radius 100 units. The other endpoints are on the surface of the sphere. We choose them so that the sphere surface is quasi uniformly sampled by them. In other words, the orientation of segments are uniformly distributed in the space. Thus we obtain a set of noise free 3D line segments in one position. Then we apply a motion, which equals $[0.4, 0.2, 0.5, 200.0, -150.0, 300.0]^t$ under the rotation axis representation, on this set and we obtain another set. Finally, independent Gaussian noise variables with zero mean and with standard deviations σ_x, σ_y and σ_z are added on the 3D coordinates of each endpoint in x, y and z directions to obtain the noisy measurements.

In the following, motion error is given in two parts: rotation error and translation error. If we use the rotation axis representation, let $\mathbf{r}$ be the real rotation parameter ($[0.4, 0.2, 0.5]^t$ in our case) and $\hat{\mathbf{r}}$ be the estimated rotation parameter, then the rotation error is defined as:
$$e_r = \|\mathbf{r} - \hat{\mathbf{r}}\| / \|\mathbf{r}\| \times 100\% \quad (40)$$
If we use the quaternion representation, we first transform it into the rotation axis representation, and then compute the error in the same way. Similarly, the translation error is defined as:
$$e_t = \|\mathbf{t} - \hat{\mathbf{t}}\| / \|\mathbf{t}\| \times 100\% \quad (41)$$
where $\mathbf{t}$ is the real translation parameter ($[200.0, -150.0, 300.0]^t$ in our case) and $\hat{\mathbf{t}}$ is the estimated one.

Since the system is nonlinear, recursive methods may give different solutions (even a wrong solution) with different initial estimate. Figures 2 and 3 show the motion errors of the four recursive methods with respect to different initial estimates. The noise level of the measurements is: $\sigma_x = \sigma_y = 2$, $\sigma_z = 6$. In **EKF-AXIS** and **EKF-QUAT**, five iterations of EKF are applied. In **MIN-AXIS** and **MIN-QUAT**, usually more than 30 iterations are needed to get a stable solution. The abscissa coordinates (from –25 to 25) correspond to different initial estimates. More precisely, the abscissa i corresponds to the initial estimate:

$[0.4\ 0.2\ 0.5\quad 200.0\ -150.0\ 300.0]^t$
$+i[0.05\ 0.05\ 0.05\ 15.00\ 15.00\ 15.00]^t$

For example, the abscissa –5 corresponds to the initial

estimate $[0.15, -0.05, 0.25, 125.0, -225.0, 225.0]^t$. The errors are the average of twenty tries. From all these results, we can say that these methods (except **MIN-QUAT**) converge in a rather wide range (from −12 to 12 for EKF methods, *i.e.*, from $[-0.2, -0.4, -0.1, 20, -330, 120]^t$ to $[1.0, 0.8, 1.1, 380, 30, 480]^t$). We find also that the methods using the quaternion representation (especially **MIN-QUAT**) is less stable than the methods using the rotation axis representation. We observe also this unstability of the quaternion representation in other experiments. Another remark is that if **MIN-AXIS** and **MIN-QUAT** do not diverge, they give exactly the same solution.

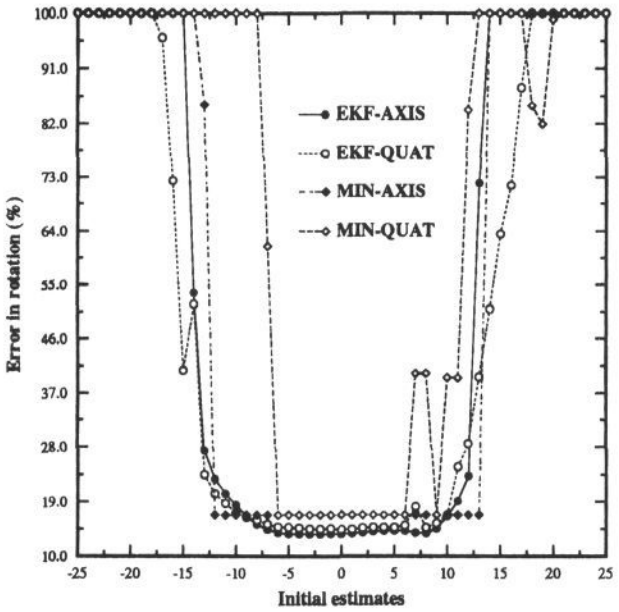

Fig. 2: ***Comparison between* EKF-AXIS, EKF-QUAT, MINAXIS *and* MIN-QUAT*: Error in rotation versus different initial estimates***

Table 2 shows the comparison on run time (on SUN 3/60 workstation), rotation error and translation error versus different methods. The results are the average of ten tries. Two line segment correspondences are used. The EKF is iterated five times. $\sigma_x = \sigma_y = 2$, $\sigma_z = 6$. From Table 2, we observe that using general minimization routine is very time expensive and that the **EIGEN** method is very efficient. EKF gives smaller motion error than other methods with a reasonable run time. This is expected since EKF takes into account the different uncertainty distribution of measurements and the others treat equally each measurement and each component of a measurement. Another remark is that using the quaternion representation is more time consuming than using the rotation axis representation. This is for two reasons:

1. The quaternion representation has one parameter more than the rotation axis representation.
2. There is a constraint for the quaternion representation and in **EKF-QUAT** we add this constraint as an additional measurement.

One can expect that the weighted least-squares method using general minimization techniques gives the best estimation, but it would take more time than **MIN-QUAT** or **MIN-AXIS**.

Figures 4 and 5 show the motion error while the standard

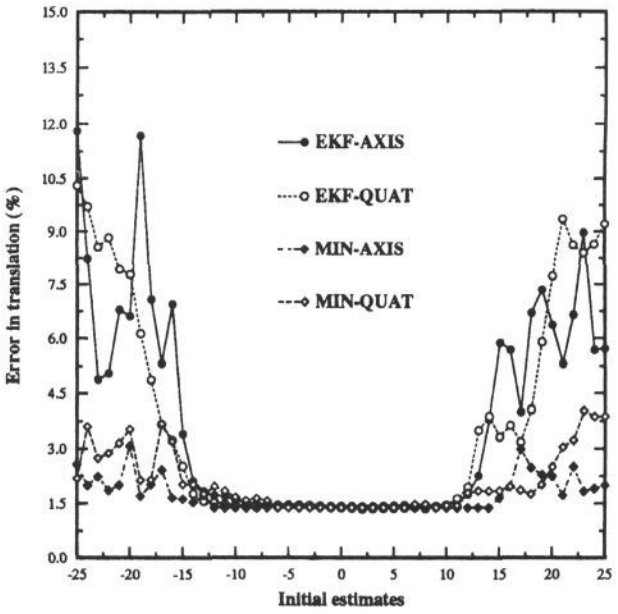

Fig. 3: ***Comparison between* EKF-AXIS, EKF-QUAT, MINAXIS *and* MIN-QUAT*: Error in translation versus different initial estimates***

Methods	*user time* (second)	*rot. error* (%)	*trans. error* (%)
MIN-QUAT	122.8	17.73	1.17
MIN-AXIS	57.7	17.73	1.17
EKF-QUAT	29.7	14.91	1.15
EKF-AXIS	27.7	14.26	1.16
EIGEN	0.07	20.72	1.12

Table 2: ***Comparison of different methods: User time and Motion error***

deviation in z direction (σ_z) varies from 1 to 20 and σ_x and σ_y are fixed at 1. The initial estimate for the recursive methods are all zero. Five iterations of EKF are applied. The error is the average of ten tries. Two correspondences are used. We can observe that error in rotation varies almost linearly with the deviation σ_z, but the slopes of the curves corresponding to **EKF-AXIS** and **EKF-QUAT** are much smaller than the others. This shows the advantage to take into account the uncertainty, especially when the uncertainty distribution is not uniform. **MIN-QUAT** gives the same error as **MIN-AXIS**, but it diverges when σ_z becomes big. From Figure 5, we see that all methods, except **EIGEN**, give almost the same error in translation.

CONCLUSION

In this paper we have presented a number of methods for determining 3D motion from line segment correspondences. Two representations of 3D line segments and two representations of rotation were described. With different representations of 3D line segment and rotation, we showed how to determine motion using the Extended Kalman Filter, a general Minimization process, and the Singular Value Decomposition. These methods were compared using both synthetic data and real data obtained by a trinocular stereo. Due to space limit, only part of the results were provided in this paper. See [14] for more

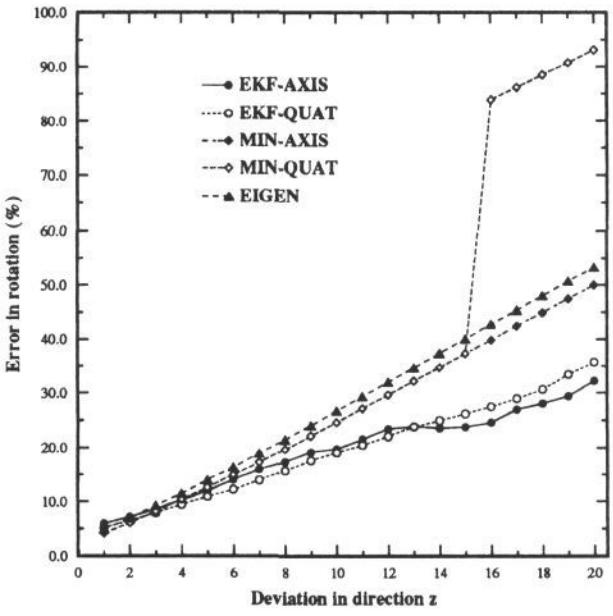

Fig. 4: Comparison: Error in rotation versus σ_z while σ_x and σ_y are fixed at 1

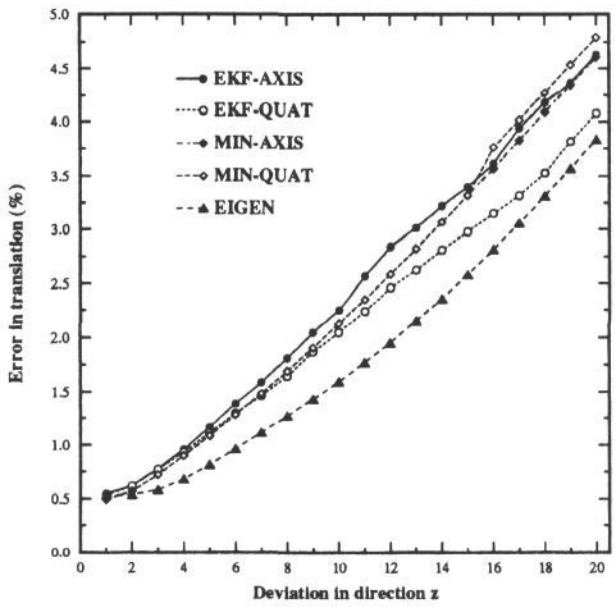

Fig. 5: Comparison: Error in translation versus σ_z while σ_x and σ_y are fixed at 1

results.

From our experiments, we observe that:

1. Uncertainty on measurements should be taken into account, especially when measurements have different uncertainty distributions.
2. When measurements have small uncertainty (less than 2% of segment length), general minimization algorithms give the best results. But when the uncertainty becomes larger, the general minimization algorithms do not give better results than EKF. Furthermore, they are more time consuming.
3. In the general minimization algorithm or EKF, using the quaternion representation is more time consuming and does not give better results than using the rotation axis representation. On the contrary, we observed that using the quaternion representation is less stable.
4. Recursive methods require an initial guess of the solution. When the initial estimate is far from the true one, recursive methods may give a wrong solution. In our experiments, we observe that the recursive methods can converge to the true solution when the initial estimate varies in a wide range from the true one.
5. Using an iterated Extended Kalman Filter can reduce the effects of non-linearity. Even when few correspondences are available, EKF converges to the true estimate after only 5 or 6 iterations.
6. Using the quaternion representation of rotation, we can use the singular value decomposition to obtain the analytical solution of motion. The methods is efficient and does not need an initial motion estimate.
7. EKF can incorporate new measurements incrementally.

We conclude that the Extended Kalman Filter with the rotation axis representation is preferable, especially when measurements have different distribution of uncertainty like in Stereo.

We note that all methods presented in this paper can be directly applied to 3D point data. In that case, at least three 3D point correspondences (non-collinear) are needed. Every three non-collinear points can form two non-parallel segments.

REFERENCES

[1] **Ayache, N.** and **Faugeras, O.D.** "Building, registrating and fusing noisy visual maps", In *Proc. First ICCV*, pages 73–82, IEEE, June 1987. London, UK.

[2] **Ayache, N.** and **Faugeras, O.D.** "Maintaining representations of the environment of a mobile robot", In *Intern. Symposium on Robotics Research*, August 1987.

[3] **Arun, K.S., Huang, T.S.** and **Blostein, S.D.** "Least-Squares Fitting of Two 3-D Point Sets", In *IEEE Trans. PAMI*, Vol.9, No. 5. Pages 698–700, September 1987.

[4] **Ayache, N.** and **Lustman, F.** "Fast and reliable passive trinocular stereovision", In *Proc. First ICCV*, pages 422–427, IEEE, June 1987. London, U.K.

[5] **Blostein, S.D.** and **Huang, T.S.** "Estimation 3-D Motion from Range Data", In *The First Conf. on Artificial Intelligence Applications*, Pages 246–250, Dec. 5–7, 1984.

[6] **Faugeras, O.D., Ayache, N.,** and **Faverjon, B.** "Building visual maps by combining noisy stereo measurements", In *Proc. Intern. Conf. on Robotics and Automation*, pages 1433–1438, April 1986. San Francisco.

[7] **Faugeras, O.D.** *A Few Steps Toward Artificial 3D Vision*, Technical Report N° 790, INRIA, 1988.

[8] **Faugeras, O.D.** and **Hebert, M.** "The Representation, Recognition, and Locating of 3D Objects", In *The International Journal of Robotics Research*, 5(3), pages 27–52, 1986.

[9] **Horn, B.K.P.** *Robot Vision*, MIT Press and McGraw–Hill Book Company, 1986.

[10] **Huang, T.S.** "Motion Analysis", In *AI Encyclopedia*, Wiley, 1986.

[11] **Maybeck, P.S.** *Stochastic Models, Estimation and Control*, Academic Press, 1979.

[12] **NAg**, *FORTRAN Library Manual* Mark 11, *Numerical Algorithms Group* 1984.

[13] **Zhang, Z., Faugeras, O.D.,** and **Ayache, N.**, "Analysis of a Sequence of Stereo Scenes Containing Multiple Moving Objects Using Rigidity Constraints", In *Proc. Second ICCV*, pages 177–186, IEEE, December, 1988, Tampa, Florida.

[14] **Zhang, Z.** *Motion Analysis from a Sequence of Stereo Frames and its Applications*, Ph.D Thesis (in English), University of Paris-Sud, 1990, to appear.