OPTIMAL MOTION AND STRUCTURE ESTIMATION

JUYANG WENG, NARENDRA AHUJA, THOMAS S. HUANG

Coordinated Science Laboratory
University of Illinois, Urbana, IL 61801

Abstract

This paper studies optimal estimation for motion and structure from point correspondences. (1) A study of the characteristics of the problem provides insight into the need for optimal estimation. (2) Methods have been developed for optimal estimation with known or unknown noise distribution. The simulations showed that the optimal estimations achieve remarkable improvement over the preliminary estimates given by the linear algorithm. (3) An approach to estimating errors in the optimized solution is presented. (4) The performance of the algorithm is compared with a theoretical lower bound — Cramér-Rao bound. Simulations show that the actual errors have essentially reached the bound. (5) A batch least-squares technique (Levenberg-Marquardt) and a sequential least-squares technique (iterated extended Kalman filtering) are analyzed and compared. The analysis and experiments show that, in general, a batch technique will perform better than a sequential technique for any nonlinear problems. Recursive batch processing technique is proposed for nonlinear problems that require recursive estimation.

1. INTRODUCTION

Although estimating motion and structure may employ long image sequences, estimating motion and structure from two perspective views is an important basic problem. The results of this basic problem provide insights into, and may be extended to, the problems that involve more images. This paper is devoted mainly to the problem of estimating motion and structure of a rigid scene from two perspective monocular views.

The optimization approach presented in this paper is motivated by the following observations of linear algorithms. (1) For certain types of motion, even pixel level perturbations (such as digitization noise) may override the information characterized by epipolar constraint. (2) Existing linear algorithms do not use the constraints in the essential parameter matrix E in solving for this matrix. These considerations are unified under a general framework of optimal estimation: Given the noise-corrupted point correspondences, we need the best estimator for motion parameters and the structure of the scene. Maximum likelihood estimation for this problem is discussed in our previous paper [Weng88b] under two types of noise distribution. However, in practice, we often do not know exact noise distribution. In this paper, we discuss optimal estimation without knowing exact noise distribution. Further, we will present approaches to estimating errors in the optimal solutions, investigate the theoretical lower bounds on the errors in the solutions and compare them with the actual errors, and analyze two types of algorithms of optimization: batch and sequential.

The remainder of the paper is organized as follows. The next section studies the need for optimization. Section 3 discusses optimal estimation without knowning noise distribution. Section 4 deals with error estimation and performance bounds. The performances of batch and sequential least-squares techniques are analyzed in Section 5. Section 6 presents experimental results. Section 7 summaries.

2. LINEAR ALGORITHMS AND THEIR STABILITY

The problem of estimating motion and structure from point correspondences of two views can be formulated as follows. Two images are taken at different positions and orientations relative to a rigid scene. The objective is to estimate the relative motion between the camera and scene as well as the structure of the scene. The coordinate system and camera model are shown in Fig. 1. Consider a feature point located at $\mathbf{x}=(x, y, z)^T$. The image plane vector of $\mathbf{x}$ in the first image is

$$\mathbf{u} = (u, v)^T = (fx/z, fy/z)^T \tag{2.1}$$

which is the perspective projection of $\mathbf{x}$ onto the image plane. Since (u, v) can be measured in terms of focal length f, without loss of generality, we assume $f=1$. Therefore, we have

$$\mathbf{u} = (u, v)^T = (x/z, y/z)^T \tag{2.2}$$

Image vector of the point $\mathbf{x}$ is defined by

$$\mathbf{X} = (u, v, 1)^T = (x/z, y/z, 1)^T \tag{2.3}$$

The motion of the scene, relative to the camera, is represented by a rotation followed by a translation. Let R be the rotation matrix and $\mathbf{T}$ be the translation vector, and let $\mathbf{x}$ move to $\mathbf{x}'$ under the motion. Then,

$$\mathbf{x}' = R\mathbf{x}+\mathbf{T} \tag{2.4}$$

Similarly, define the image plane vector $\mathbf{u}'$ of the image vector $\mathbf{X}'$:

$$\mathbf{X}' = (u', v', 1)^T = (x'/z', y'/z', 1)^T \tag{2.5}$$

2.1 Linear Algorithms and Epipolar Constraint

The linear algorithms published in literature share the same key structure: determining intermediate parameters, called essential parameters, based on epipolar constraint followed from (2.4):

$$(\mathbf{X}')^T(\mathbf{T}\times(R\,\mathbf{X})) = 0 \tag{2.6}$$

where $\times$ denotes vector cross product. That is, $\mathbf{X}'$, $R\mathbf{X}$ and $\mathbf{T}$ are coplanar. Its geometrical illustration is shown in Fig. 2. We define a mapping $[\cdot]_\times$ from a 3-dimensional vector to a 3 by 3 matrix:

$$[(x_1, x_2, x_3)^T]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \tag{2.7}$$

CH2752-4/89/0000/0144$01.00 © 1989 IEEE

Using this mapping, we can express cross product of two vectors by the matrix multiplication of a 3 by 3 matrix and a column matrix:

$$\mathbf{X}\times\mathbf{Y} = [\mathbf{X}]_\times\mathbf{Y} \tag{2.8}$$

Define E to be

$$E = [\mathbf{T}_s]_\times R \tag{2.9}$$

where $\mathbf{T}_s$ is a unit vector such that $\mathbf{T}\times\mathbf{T}_s=\mathbf{0}$. (2.6) can be rewritten as

$$(\mathbf{X}')^T E\,\mathbf{X} = 0 \tag{2.10}$$

(2.10) is a linear equation in the elements of matrix E. Using 8 or more point correspondences, the linear algorithms first solve for E based on (2.10) and then solve for motion parameters from E.

It is ready to see that only one component of the image position of a point (the one perpendicular to the epipolar line) is used by epipolar constraint: The vectors $\mathbf{X}'$ and $R\,\mathbf{X}$ can be perturbed in the plane where $\mathbf{X}'$, $R\,\mathbf{X}$ and $\mathbf{T}$ lie without violating (2.10). In other words, the location of the points on the epipolar line is irrelevant to epipolar constraint. It is related to the depth of the point as well as motion parameters. The questions to ask are: (1) The essential matrix E has only 5 degrees of freedom (2 for unit vector $\mathbf{T}_s$ and 3 for rotation matrix R). How can the constraint in E be used to improve accuracy in the presence of noise? (2) How reliably the motion parameters can be estimated using the epipolar constraint? (3) Can another component of the point (along the epipolar line) be used, in addition to the epipolar constraint, to improve the reliability of the estimated motion parameters and structure in the presence of noise? These problems are investigated in the following.

2.2 Using the Constraint in the Essential Matrix E

By definition of Equation (2.9), E should be decomposable into the product of a skew symmetric matrix ($S=-S^T$) and a rotation matrix R (orthonormal with determinant 1). The following theorem states a necessary and sufficient condition for a matrix E to satisfy the definition of (2.9).

Theorem 1. Given a 3 by 3 matrix E, the necessary and sufficient condition for existing a rotation matrix R and a skew symmetric matrix S ($S^T=-S$), such that $E=SR$, is that one of the eigenvalues of E^TE is equal to 0 and the other two are equal.

Proof. See Appendix.

Corollary. Given a 3 by 3 matrix E, the necessary and sufficient condition for existing a rotation matrix R and a unit vector $\mathbf{T}_s$, such that $E=[\mathbf{T}_s]_\times R$, is that the eigenvalues of E^TE are 0, 1, 1, respectively.

Proof. See Appendix.

The constraint on the eigenvalues of E, stated in Theorem 1, can be written as polynomial equations in the elements of E. These yield more equations, in addition to (2.10). In the case where 8 or more points are available, E can be solved using linear equations (2.10) without considering those nonlinear constraints in E. However, in the presence of noise, E solved from the linear equations generally does not satisfy the condition in Theorem 1. The constraint in E can be used by iteratively improving some "independent parameters" in R and $\mathbf{T}_s$ to minimize the weighted sum of $((\mathbf{X}')^T(\mathbf{T}_s\times R\,\mathbf{X}))^2$:

$$\sum_{i=1}^{n}\frac{((\mathbf{X}_i')^T(\mathbf{T}_s\times R\,\mathbf{X}_i))^2}{\sigma^2(\|R^T(\mathbf{T}_s\times\mathbf{X}_i')\|_{z=0}^2+\|\mathbf{T}_s\times R\,\mathbf{X}_i\|_{z=0}^2)} \tag{2.11}$$

where $\|(a,b,c)\|_{z=0}^2 = a^2+b^2$ and each denominator is the variance of the first order perturbation of $(\mathbf{X}')^T(\mathbf{T}_s\times R\,\mathbf{X})$, assuming the components of $\mathbf{u}$ and $\mathbf{u}'$ have additive uncorrelated zero mean noise with variance σ^2. We call this minimizing epipolar errors or epipolar improvement.

2.3 A Type of Motion

Let us consider a type of pure translation and the corresponding type of pure rotation. For the type of translation, the translation vector and the optical axis is orthogonal. For the type of pure rotation the rotation axis is orthogonal to the optical axis and the translation vector of the type of pure translation. Without loss of generality, let the translation direction be parallel to Y axis, the rotation axis is parallel to X axis. Fig. 3 shows examples of the displacement fields of the pure translation and the pure rotation, respectively. It is clear that the translation produces horizontal displacement vectors and the rotation produces almost horizontal ones. We analyze this property quantitatively. For horizontal pure translation $\mathbf{T}=(0, t_2, 0)^T$, from $\mathbf{x}'=\mathbf{x}+\mathbf{T}$ it is easy to see that the left side of (2.6) is

$$(\mathbf{X}')^T(\mathbf{T}\times(R\,\mathbf{X})) = u'-u \tag{2.12}$$

which should be equal to 0. Therefore the epipolar constraint for this pure horizontal translation is that the image plane displacement vector $\mathbf{u}'-\mathbf{u}$ is horizontal.

For the pure rotation, the rotation matrix is given by

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{2.13}$$

From $\mathbf{x}'=R\,\mathbf{x}$ it follows that

$$u'-u = \frac{u}{v\sin\theta+\cos\theta}-u = \frac{u(1-\cos\theta-v\sin\theta)}{v\sin\theta+\cos\theta} \tag{2.14}$$

$$v'-v = \frac{v\cos\theta-\sin\theta}{v\sin\theta+\cos\theta}-v = -\frac{(1+v^2)\sin\theta}{v\sin\theta+\cos\theta} \tag{2.15}$$

Since generally $u'-u\neq 0$, the pure rotation does not exactly satisfy the epipolar constraint of the horizontal translation. However, the value of $u'-u$ is almost equal to zero. Assume the image size is s (image is a s by s square) with $m\times m$ pixels. The pixel size is then s/m. The vertical displacement in (2.14) in terms of the number of pixels is then

$$(u'-u)m/s = \frac{mu(1-\cos\theta-v\sin\theta)}{s(v\sin\theta+\cos\theta)} \tag{2.16}$$

Fig. 4 shows the value of the vertical displacement in (2.16), in terms of number of pixels, at the center of a quadrant of an image with 512 by 512 resolution, for different horizontal displacement, $v'-v$, and different image sizes, s. Since the vertical displacement is very small for the rotation, the corresponding displacement field almost satisfies the epipolar constraint for the pure translation discussed earlier. This implies that in the presence of small errors in the image coordinates (e.g., in the magnitude of one or two pixels), pure translation can be interpreted by a rotation as far as the epipolar constraint is concerned. In other words, the epipolar constraint cannot disambiguate the translation from the corresponding rotation in the presence of even small image digitization errors. Furthermore, we can show that the number of pixels of vertical displacement goes to zero quadratically as the image size s approaches zero. This implies that motion estimation based on epipolar constraint is extremely unreliable for small field of view.

2.4 Beyond the Epipolar Constraint

From (2.8) it can be seen that the depths of the object points are excluded in epipolar constraint. This is desirable to the linear algorithms since the depths of the points are unknown. The epipolar constraint uses only one component (perpendicular to the epipolar line) of image points. However, the other component (parallel

to the epipolar line) left by epipolar constraint is important for determining motion parameters from the following properties: (1) Under a rotation, all the points on a projection line (passing through the origin and an image point) project on to a common image point after rotation. (2) Under a translation, those points on a projection line have different projections after rotation: the closer the point is to the image plane, the larger the displacement of the point is in the image plane. Fig. 5 illustrates these two properties. These properties can be proved analytically and the proof is omitted here. As long as $1/z'$ has a large variation among points, the displacement field is quite different between a translation and a rotation. For example, in Fig. 3 the image displacement vectors are almost horizontal for both rotation and translation. However, the lengths of the displacement vectors are quite different for translation, but are similar for rotation. It is impossible to interpret a translation by a rotation if both components of the points are used. The optimization discussed later in the paper makes use of both components of the image points. As shown by simulations, this significantly improves reliability of the solutions.

3. OPTIMAL ESTIMATION

The observed 2-D image coordinate vectors $\mathbf{u}_i$ of image 1 and $\mathbf{u}'_i$ of image 2 are noise corrupted versions of the true image vectors. Therefore $(\mathbf{u}_i, \mathbf{u}'_i)$ is the observed value of a pair of random vectors $(\mathbf{U}_i, \mathbf{U}'_i)$. (With n point correspondences over two time instants, we add subscripts i to denote the ith point and the subscript-free letters denote the general collection of vectors.) What we obtain is a sequence of the observed image vector pairs

$$\mathbf{u} \triangleq (\mathbf{u}_1^T, (\mathbf{u}'_1)^T, \mathbf{u}_2^T, (\mathbf{u}'_2)^T, \cdots, \mathbf{u}_n^T, (\mathbf{u}'_n)^T)^T \tag{3.1}$$

of a sequence of random vector pairs

$$\mathbf{U} \triangleq (\mathbf{U}_1^T, (\mathbf{U}'_1)^T, \mathbf{U}_2^T, (\mathbf{U}'_2)^T, \cdots, \mathbf{U}_n^T, (\mathbf{U}'_n)^T)^T \tag{3.2}$$

We need to estimate the motion parameters $\mathbf{M}$ and the 3-D positions of the feature points (scene structure)

$$\mathbf{X} \triangleq (\mathbf{x}_1^T, (\mathbf{x}'_1)^T, \mathbf{x}_2^T, (\mathbf{x}'_2)^T, \cdots, \mathbf{x}_n^T, (\mathbf{x}'_n)^T)^T \tag{3.3}$$

If distribution of noise in the observations are known, maximum likelihood estimator is an appropriate optimal estimator. Two types of noise distribution has been discussed in [Weng88b]: Gaussian distribution and the distribution of the so called uncertainty polyhedron. If the image coordinates of points are corrupted by white Gaussian noise, the problem of maximum likelihood estimator of motion parameter vector $\mathbf{m}$ is such that *image error*,

$$\|\mathbf{u}-\mathbf{h}(\mathbf{m}, \mathbf{x})\|/\sqrt{2n} \tag{3.4}$$

is minimized, where $\mathbf{h}(\mathbf{m}, \mathbf{x})$ is the computed noise-free version of $\mathbf{u}$ based on motion parameter $\mathbf{m}$ and structure $\mathbf{x}$. In other words, the discrepancies between the observations and the inferred projections are minimized. Since the optimal structure $\mathbf{x}$ can be computed from $\mathbf{u}$ and $\mathbf{m}$ [Weng88b], for simplicity we just use $\mathbf{m}$ to denote the unknown parameters to be estimated. The solution of a linear algorithm (e.g., [Weng87]) is used to provide an initial guess solution for the nonlinear minimization algorithm.

However, the distribution of noise is usually unknown. We now discuss optimal estimator with unknown error distribution. Let $\mathbf{m}$ be the parameters to be estimated and $\hat{\mathbf{m}}(\mathbf{u})$ be the estimate based on the observation vector $\mathbf{u}$. The error vector of $\hat{\mathbf{m}}$ is $\delta_\mathbf{m} \triangleq \hat{\mathbf{m}}-\mathbf{m}$. Given the parameter $\mathbf{m}$, the estimated parameter $\hat{\mathbf{m}}$ is random due to random noise in the observations $\mathbf{u}$. Two widely used criteria are that the estimate is unbiased

$$\mathbf{E}\hat{\mathbf{m}} = \mathbf{m} \tag{3.5}$$

and it minimizes the expected squared norm of the error vector

$$\mathbf{E}\|\delta_\mathbf{m}\|^2 = \mathbf{E}\|\hat{\mathbf{m}}-\mathbf{m}\|^2 \tag{3.6}$$

The estimate that minimizes (3.6) is called minimum variance estimate (or least-squares estimate, minimum mean square estimate). Let us first consider a linear least-squares problem.

Gauss-Markov Theorem. Suppose

$$\mathbf{y} = A\mathbf{m}+\delta_\mathbf{y} \tag{3.7}$$

where $\delta_\mathbf{y}$ is a random vector with zero mean, $\mathbf{E}\delta_\mathbf{y}=0$, and covariance matrix

$$\Gamma_\mathbf{y} = \mathbf{E}\delta_\mathbf{y}\delta_\mathbf{y}^T \tag{3.8}$$

The unbiased, linear minimum variance estimator of $\mathbf{m}$ that minimizes $\mathbf{E}\|\hat{\mathbf{m}}-\mathbf{m}\|^2$ is

$$\hat{\mathbf{m}} = (A^T\Gamma_\mathbf{y}^{-1}A)^{-1}A^T\Gamma_\mathbf{y}^{-1}\mathbf{y} \tag{3.9}$$

with error covariance matrix

$$\Gamma_{\hat{\mathbf{m}}} \triangleq \mathbf{E}(\hat{\mathbf{m}}-\mathbf{m})(\hat{\mathbf{m}}-\mathbf{m})^T = (A^T\Gamma_\mathbf{y}^{-1}A)^{-1} \tag{3.10}$$

For proofs, see, e.g., [Luen69] [Sore80] [Gior85]. The estimator corresponds to the least-squares estimator with weight matrix $\Gamma_\mathbf{y}^{-1}$, i.e., it minimizes

$$(\mathbf{y}-A\mathbf{m})^T\Gamma_\mathbf{y}^{-1}(\mathbf{y}-A\mathbf{m}) \tag{3.11}$$

For white noise, $\Gamma_\mathbf{y}=\sigma I$, By Gauss-Markov Theorem we know that the minimum variance estimator of $\mathbf{m}$ is the one that minimizes $\|\mathbf{y}-A\mathbf{m}\|$.

Consider again the maximum likelihood estimator for white Gaussian noise distribution. The maximum likelihood estimator of $\mathbf{m}$ minimizes the norm of $\mathbf{f}(\mathbf{m}) \triangleq \mathbf{u}-\mathbf{h}(\mathbf{m})$. Given an estimated $\mathbf{m}$, $\mathbf{m}_i$, expanding $\mathbf{f}(\mathbf{m})$ at $\mathbf{m}_i$ yields

$$\mathbf{f}(\mathbf{m}) \approx \mathbf{f}(\mathbf{m}_i)+J_i\ (\mathbf{m}-\mathbf{m}_i)+o(\|\mathbf{m}-\mathbf{m}_i\|) \tag{3.12}$$

where

$$J_i \triangleq \frac{\partial \mathbf{f}(\mathbf{m}_i)}{\partial \mathbf{m}} \tag{3.13}$$

or

$$-\mathbf{f}(\mathbf{m}_i)+J_i\mathbf{m}_i = J_i\mathbf{m}-\mathbf{f}(\mathbf{m})+o(\|\mathbf{m}-\mathbf{m}_i\|) \tag{3.14}$$

Let

$$\mathbf{y}_i = -\mathbf{f}(\mathbf{m}_i)+J_i\mathbf{m}_i$$

we have

$$\mathbf{y}_i = J_i\mathbf{m}-\mathbf{f}(\mathbf{m})+o(\|\mathbf{m}-\mathbf{m}_i\|) \tag{3.15}$$

Neglecting the higher order term $o(\|\mathbf{m}-\mathbf{m}_i\|)$, (3.15) is of the form of (3.7). Instead of assuming independent Gaussian noise, we just assume that the noises are uncorrelated and have the zero mean and equal variance. By the Gauss-Markov Theorem, we need to minimize

$$\|\mathbf{f}(\mathbf{m})\| \approx \|\mathbf{y}_i - J_i\mathbf{m}\| \tag{3.16}$$

which leads

$$\mathbf{m} = \mathbf{m}_i-(J_i^TJ_i)^{-1}J_i^T\mathbf{f}(\mathbf{m}_i) \tag{3.17}$$

Since $J_i^TJ_i$ may be singular, $J_i^TJ_i$ is replaced by $D_i+J_i^TJ_i$, where D_i is a diagonal matrix with non-negative diagonal elements. Then we get a sequence of approximation to a minimum point of (3.16)

$$\mathbf{m}_{i+1} = \mathbf{m}_i+(D_i+J_i^TJ_i)^{-1}J_i^T\mathbf{f}(\mathbf{m}_i) \tag{3.18}$$

This is called Levenberg-Marquardt or L-M method [Leve44] [Marq63] [Luen82]. The special case in which $D_i=0$ for all i is called Gauss', or the Gauss-Newton method [Luen82]. Each iteration leads to an unbiased linear minimum variance estimator based on the locally linearized equation of (3.15), neglecting the higher order terms. When iteration converges, if the converged point $\mathbf{m}_i$

is not far from the true solution, $o(\|\mathbf{m}-\mathbf{m}_i\|)$ is small, (3.15) is a good approximation of the system. If the convergence occurs far from the true solution (e.g., the iteration is trapped at a local minimum, or the noise is extremely large) $o(\|\mathbf{m}-\mathbf{m}_i\|)$ is large and the linearized model does not well characterize the system.

In summary, if the locally linearized model is considered the unbiased linear minimum variance estimator also leads to minimizing the norm of $\mathbf{f}(\mathbf{m})$. Therefore, the objective of minimizing $\|\mathbf{f}(\mathbf{m})\|$ can be used for general white noise. If the image plane noise are colored, the regular norm of $\mathbf{f}(\mathbf{m})$ should be replaced by a weighted norm according to the Gauss-Markov theorem.

4. ERROR ESTIMATION AND PERFORMANCE BOUNDS

We need to investigate the following two issues. (1) How can we access the reliability of the solutions? (2) What is the theoretical bound of the performance and how close are the errors of the algorithm to the bound? These two questions are investigated in the following subsections. Since the errors in the measurements are random, accuracy or performance of the estimator (or algorithm) should be investigated in statistical sense.

4.1 Error Estimation

Error estimation for linear algorithms has been studied in [Weng87], where the key is basically to estimate errors in the least-squares solution of a linear system $A\mathbf{m}=\mathbf{y}$, where both matrix A and $\mathbf{y}$ are corrupted by noise. The problem here is different since only $\mathbf{y}$ is corrupted by noise (see (3.15)) and computation is iterative. The estimated error indicates the reliability of the solution, which depends on not only noise level, but also structure of the scene, motion parameters and the parameters of system [Weng89b].

The minimum variance estimation discussed above leads to a method for estimating errors in the estimates. By Gauss-Markov Theorem, the covariance matrix of the error vector $\hat{\mathbf{m}}-\mathbf{m}$ of a linear problem is give by (3.10) For the nonlinear problem investigated here, the matrix A corresponds to

$$J = \frac{\partial \mathbf{f}(\mathbf{m})}{\partial \mathbf{m}} \tag{4.1}$$

evaluated at the finally estimated parameter $\hat{\mathbf{m}}$. For uncorrelated uniform variance noise, $\Gamma_{\mathbf{y}}=\sigma^2 I$, the covariance matrix is simply

$$\Gamma_{\hat{\mathbf{m}}} = \mathbf{E}(\hat{\mathbf{m}}-\mathbf{m})(\hat{\mathbf{m}}-\mathbf{m})^T = \sigma^2(J^TJ)^{-1} \tag{4.2}$$

The trace of the covariance matrix gives the expected squared norm of error vector

$$trace\,(\Gamma_{\hat{\mathbf{m}}}) = \mathbf{E}(\hat{\mathbf{m}}-\mathbf{m})^T(\hat{\mathbf{m}}-\mathbf{m}) = \mathbf{E}\,\|\hat{\mathbf{m}}-\mathbf{m}\|^2 \tag{4.3}$$

Since the optimization established in the last section just assume zero-mean and covariance matrix of the noise, the error estimation discussed here does not require specific noise distribution either.

We have implemented both analytical and finite difference versions for the partial derivatives in J. The estimated errors show very little difference between those two versions. In practice, the finite difference version is easier to programming.

Motion parameters can be represented in many ways. Sometimes it is necessary to know the errors in terms of the required representation. Generally, for a representation $\mathbf{m}'=\mathbf{g}(\mathbf{m})$ which is a function of the parameter $\mathbf{m}$. We have

$$\hat{\mathbf{m}}'-\mathbf{m}' \cong \frac{\partial \mathbf{g}(\hat{\mathbf{m}})}{\partial \mathbf{m}}(\mathbf{m}-\hat{\mathbf{m}})$$

Therefore the covariance matrix of the error vector of $\hat{\mathbf{m}}'$ can be estimated by

$$\Gamma_{\hat{\mathbf{m}}'} = \mathbf{E}(\hat{\mathbf{m}}'-\mathbf{m}')(\hat{\mathbf{m}}'-\mathbf{m}')^T = \frac{\partial \mathbf{g}(\hat{\mathbf{m}})}{\partial \mathbf{m}}\Gamma_{\hat{\mathbf{m}}}\left(\frac{\partial \mathbf{g}(\hat{\mathbf{m}})}{\partial \mathbf{m}}\right)^T$$

4.2 Performance Bounds

There exist theoretical bounds for the covariance matrix of any estimator. Cramér-Rao bound is one of them [Cram46] [Rao75]. The application of the bound to the problem discussed here is presented in [Weng89a]. Let $\mathbf{f}(\mathbf{m})=\mathbf{u}-\mathbf{h}(\mathbf{m})$ is a white Gaussian noise vector with variance σ, then for the unbiased estimator $\hat{\mathbf{m}}$, the Cramér-Rao bound gives

$$\Gamma_{\hat{\mathbf{m}}} \le F^{-1} = \sigma^2(J^TJ)^{-1} \tag{4.4}$$

assuming J^TJ has a full rank.

It is interesting to recall error estimation discussed in Section 4.1. The estimated error in (4.2) takes the same form as the bound in (4.4). However, they are different. J in (4.2) is evaluated with the estimated $\mathbf{m}$ and noise corrupted observation $\mathbf{u}$ while J in (4.4) is evaluated with the true $\mathbf{m}$ and noise-free observation. It is important to note that the bound is independent of any algorithms. (The expression of the Cramér-Rao bound is used in [Broi86] to estimate the errors of the corresponding algorithms.) For more discussions of the Cramér-Rao bound and Bhattacharyya bound for motion analysis, see [Weng89a] where simulations show that for the optimized solution, the actual bias is small and the actual errors are very close to the Cramér-Rao bound for unbiased estimators. In other words, the errors are very close to those that would result from the "best possible" unbiased estimator.

5. BATCH, SEQUENTIAL, AND RECURSIVE BATCH SOLUTIONS

If all the data acquired are processed simultaneously, the corresponding processing method is called batch processing. If a new solution is computed after each set of new data is acquired, and the new solution is computed based on the old solution and the new data, the method is called sequential processing. Due to the popularity of a sequential processing technique called Kalman filter, sequential processing method has been used for many applications. Therefore, it is very important to analyze and compare the performances of batch techniques and sequential techniques. In fact, although the solutions of both types of techniques are mathematically the same for linear problems, the solution of the batch techniques is generally better than that of the sequential techniques for nonlinear problems. For the cases where solutions are needed while new data are collected, the so called *recursive batch technique* is proposed to improve the performance for nonliner problems.

Although the above conclusions apply to time-varying systems (parameters vary with time) we will only briefly discuss time invariant systems due to limit of space. For the linear system in (3.7), Gauss-Markov theorem provides an unbiased linear minimum variance estimator of $\mathbf{m}$. The estimate can also be computed sequentially by Kalman filter technique. Kalman algorithm can be derived by either probabilistic or deterministic methods. Both types methods are unified under Hilbert space optimization [Luen69] [Gior85]. It can be proved that Kalman filter gives the same estimate for a linear problem (starting with ∞I as an initial covariance matrix). However, for a nonlinear problem, the results given by a batch solution are different from those given by the corresponding Kalman filter (called iterated extended Kalman filter,

or IEKF). For a batch approach, the solution $\hat{\mathbf{m}}$ that minimizes $\|\mathbf{f}(\hat{\mathbf{m}})\|^2=\sum_{j=1}^{n}\|\mathbf{f}_j(\hat{\mathbf{m}})\|^2$ is computed recursively to minimize (see (3.16)):

$$\sum_{j=1}^{n}\|\mathbf{f}_j(\hat{\mathbf{m}})\|^2 \approx \sum_{j=1}^{n}\|\mathbf{y}_j - J_j(\hat{\mathbf{m}})\,\hat{\mathbf{m}}\|^2 \tag{5.1}$$

where, index j denotes the corresponding variable for jth 3-D point. However, the Kalman filter minimizes instead

$$\sum_{j=1}^{n}\|\mathbf{y}_j - J_j(\hat{\mathbf{m}}^{(j)})\,\hat{\mathbf{m}}\|^2 \tag{5.2}$$

where $\hat{\mathbf{m}}^{(j)}$ is the estimated $\mathbf{m}$ based on first j points. The key difference is the point where the matrix J_j is evaluated. By the batch approach the matrices $\{J_j\}$ are always updated when new estimates of $\mathbf{m}$ is computed in iteration. By the sequential approach (IEKF), the matrix is not updated when new observations are collected ($J_j(\hat{\mathbf{m}}^{(j)})$ is based on only first j points). For small j, $\hat{\mathbf{m}}^{(j)}$ is very poor since just j observations are available. Therefore J_j evaluated at $\hat{\mathbf{m}}^{(j)}$ gives a system matrix that is evaluated far from the true parameter. This results in inaccurate system equations. Once those inaccurate system equations are generated, they will not be changed later. They are included in the objective function (5.2), further preventing the estimated parameter $\hat{\mathbf{m}}$ from approaching the correct parameters while new data are obtained. For the same reason, the error covariance matrix of $\mathbf{m}$ computed by IEKF underestimates the errors. Therefore, sequential methods generally are not suitable to be used as a tool to solve a very nonlinear equation from an arbitrary initial guess for $\mathbf{m}$.

For the cases where estimates are required while data are collected, or the number of observations is extremely large, batch methods are computationally expensive since repeated optimization covers all previous data. However, very old data contribute very little to the current estimate. If the updating for the very old data is neglected the batch techniques can be used for recent l observations. This is the key idea of what we called *recursive batch technique*. l should be sufficiently large such that updating for observations older than recent l observations can be neglected.

6. EXPERIMENTAL RESULTS

To further verify the analyses presented above, simulations have been performed. The algorithm has also tested on images of real world scenes.

For the simulations, the focal length of the simulated camera is one. The image frame is a $s\times s$ square. Unless stated otherwise, s=0.70, and 12 point correspondences are used. The object points are generated randomly with depth ranging between 5 to 16. Only those points that are visible before and after motion are used. Noise is measured in terms of resolution: the variance of the noise is equal to that of the pixel round-off noise. All errors shown in this section are relative. Relative error of a matrix, or vector, is defined by the Euclidean norm of the error matrix (square root of the sum of squared elements), or vector, divided by the Euclidean norm of the correct matrix, or vector, respectively.

1. **Minimizing Epipolar Errors Versus Minimizing Image Errors**. The errors after the epipolar improvement is compared with that of minimizing image errors (both using Levenberg-Marquardt method). For longitudinal translation (parallel to optical axis), both methods give very similar errors, since the solution of the linear algorithm are already fairly good (see [Weng89b]). The performance shows difference for unstable lateral translations (parallel to image plane). Fig. 6 shows typical average errors over 100 random trials. It can be seen that the epipolar improvement is significant compared with linear algorithm. However, the errors after minimizing image errors are further significantly smaller than those after epipolar improvement, especially for a small field of view. Therefore the use of two components in image point coordinates does significantly improve the robustness.

2. **Sequential Versus Batch Solutions**. We obtain batch solution using Levenberg-Marquardt method for nonlinear least-squares problems. The sequential solution is obtained by iterated extended Kalman filter (IEKF). Iteration is performed for a point correspondence (4 observations: 2 components for each of two images) to improve the performance of regular IEKF (which just iterates on one observation). Such iterations are performed for each point correspondence until no improvement for the parameters occurs. With a zero rotation angle, an arbitrary translation direction and a large covariance matrix, IEKF always diverges. The solution given by IEKF has more than 100% relative errors. With a good initial guess provided by the linear algorithm, the solutions of IEKF are quite different according to different initial covariance matrix $P_{0,-1}$. If the diagonal matrix $P_{0,-1}$ are assigned large diagonal values, IEKF may diverge. Only with small diagonal values, IEKF can improve the initial guesses. Fig. 7 shows typical sequences of Kalman sequential updating. This example confirms that Kalman filter does not perform well with arbitrary initial guess, and the error variance given by the Kalman algorithm is misleading. The performances of the linear algorithm, a sequential algorithm (IEKF) and a batch algorithm (Levenberg-Marquardt method, or L-M method) are compared in Fig. 8. Both IEKF and L-M method use the solutions of the linear algorithm as initial guesses. The initial covariance matrix of IEKF is optimized to be relatively small (same as that in Fig. 7(a)). The image errors in Fig. 8(d) show that the batch solution find the minimum very consistently. However, IEKF does not always find the minimum. Fig. 8 confirms that batch optimization performs significantly better than IEKF sequential algorithm. Other simulations indicated that the average image errors by a batch approach are always close to average errors in the image coordinates including extremely noisy cases: 32×32 pixels.

3. **Error Estimation and Cramér-Rao Bound**. Fig. 9 shows the average relative errors, average deviation of the error estimation (the absolute difference between the estimated relative errors and the actual relative errors) and the bias of error estimation (average difference between the estimated relative errors and the actual relative errors) over 40 random trials. As can be seen from the figure, the average deviation is generally less than a half of the magnitude of the actual relative errors. The bias is also small. The results concerning essential reaching Cramér-Rao bound are shown in [Weng89a].

4. **Real World Images**. A CCD video camera with roughly 480×500 pixels is used as image sensor. The focal length of the camera is calibrated but no nonlinearity correction is made for the camera. The camera takes two images at different positions. A two-view matcher computes image displacement field and occlusion on a pixel grid [Weng88a]. Samples of the displacement field computed for a scene (Mac scene) are shown on a 13 by 14 grid in Fig. 10(a), superimposed on the first image, which is extended to provide context for the peripheral areas of the image. Those 13×14=182 displacement vectors shown in Fig. 10(a) are used as point correspondences to compute motion parameters. The motion parameters computered are shown in Table 1.a. As shown in the table, the image error is within a half pixel width. The plot of depth map is shown in Fig. 10(b). Assume the errors in the coordinates of the matching points are uncorrelated. The estimated variance of the errors are given by the squared image errors. The estimated errors of the Mac Scene are shown in Table 1.b. Fig. 11 shows the samples of the computed displacement filed of a path

scene. The data and results are shown in Table 2.a. The estimated errors are shown in Table 2.b.

7. SUMMARY

The paper first discusses a type of motion for which the algorithms based on the epipolar constraint are very sensitive to noise. The analysis and simulations lead to the conclusion that it is important to use both components of image positions of points in determining motion parameters in the presence of noise. Optimal estimation of motion and structure parameters is investigated for known and unknown noise distributions. An approach to estimating errors in the optimal estimates is introduced and implemented. Analysis and simulations also lead to the flowing conclusions: For nonlinear problems, the performance of IEKF algorithm is inferior to that of Levenberg-Marquardt algorithm, and the covariance matrices given by IEKF may significantly underestimate the variance of errors. A recursive-batch processing approach is proposed for those problems where estimates are required sequentially while observations are collected.

APPENDIX

Theorem 1. Given a 3 by 3 matrix E, the necessary and sufficient condition for existing a rotation matrix R and a skew symmetric matrix S ($S^T=-S$), such that $E=SR$, is that one of the eigenvalues of E^TE is equal to 0 and the other two are equal.

Proof. By our notation of $[\mathbf{T}]_\times$, the following is a trivial fact: Given any 3×3 skew symmetric matrix $S=[s_{ij}]$ letting $\mathbf{T}=(-s_{23}, s_{13}, -s_{12})^T$, we have $[\mathbf{T}]_\times=S$. Conversely given a 3-dimensional vector $\mathbf{T}$, $[\mathbf{T}]_\times$ is a skew symmetric matrix. Therefore, under the mapping of $[\mathbf{T}]_\times$ there exist a one-to-one correspondence between all the 3×3 skew symmetric matrices and all three-dimensional vectors.

Necessity part: Let $E=[\mathbf{T}]_\times R$, where R is a rotation matrix. If $\mathbf{T}=\mathbf{0}$, $E=\mathbf{0}$ and all three eigenvalues of E^TE are equal to 0, and the necessity is true. Now assume $\mathbf{T}\neq\mathbf{0}$. Define an orthonormal matrix

$$H = R^{-1}[\hat{\mathbf{T}}, \hat{\mathbf{T}}_2, \hat{\mathbf{T}}_3] \qquad \text{(A.1.1)}$$

where $\hat{\mathbf{T}}=\mathbf{T}/\|\mathbf{T}\|$, and $\hat{\mathbf{T}}_2, \hat{\mathbf{T}}_3$ are such that $\hat{\mathbf{T}}, \hat{\mathbf{T}}_2, \hat{\mathbf{T}}_3$ are orthonormal. We have

$$EH = [\mathbf{T}]_\times[\hat{\mathbf{T}}, \hat{\mathbf{T}}_2, \hat{\mathbf{T}}_3] = [\mathbf{0}, \mathbf{V}_2, \mathbf{V}_3] \qquad \text{(A.1.2)}$$

where $\mathbf{V}_i = \mathbf{T}\times\hat{\mathbf{T}}_i$, $i=2,3$. Obviously $\mathbf{V}_2$ and $\mathbf{V}_3$ are orthogonal and $\|\mathbf{V}_i\|=\|\mathbf{T}\|$, $i=2,3$. Therefore,

$$H^tE^tEH = diag(0, \|\mathbf{T}\|^2, \|\mathbf{T}\|^2) \qquad \text{(A.1.3)}$$

Sufficiency part: Suppose E^TE has three eigenvalues 0, λ, λ. Then there exists a rotation matrix H such that

$$H^tE^tEH = diag(0, \lambda, \lambda) \qquad \text{(A.1.4)}$$

(A.1.4) implies that the first column of EH is a zero vector and the remaining two columns are orthogonal with length $\sqrt{\lambda}$. Therefore there exist a rotation matrix Q such that

$$EH = Qdiag(0, \sqrt{\lambda}, \sqrt{\lambda}) \qquad \text{(A.1.5)}$$

On the other hand,

$$diag(0, \sqrt{\lambda}, \sqrt{\lambda}) = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\sqrt{\lambda} \\ 0 & \sqrt{\lambda} & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \triangleq SR \qquad \text{(A.1.6)}$$

(The use of this seemingly odd equation was guided by our original proof which is more natural but less compact). From (A.1.5) and (A.1.6), we get

$$E = Qdiag(0, \sqrt{\lambda}, \sqrt{\lambda})H^T = QSRH^T = \{QSQ^T\}\{QRH^T\} \qquad \text{(A.1.7)}$$

where in the last expression, the first part is a skew symmetric matrix and the second part is a rotation matrix. This is to be proved.

Corollary. Given a 3 by 3 matrix E, the necessary and sufficient condition for existing a rotation matrix R and a unit vector $\hat{\mathbf{T}}$, such that $E=[\hat{\mathbf{T}}]_\times R$, is that the eigenvalues of E^TE are 0, 1, 1, respectively.

Proof. Procedure of the proof is very similar to the above. For necessity part let $\|\mathbf{T}\|=1$. For sufficiency part $\lambda=1$. Let $[\hat{\mathbf{T}}]_\times=QSQ^T$.

$$2\|\hat{\mathbf{T}}\|^2 = trace(QSQ^T)(QSQ^T)^T = trace(QSS^TQ^T) = trace(SS^T) = 2$$

Therefore $\hat{\mathbf{T}}$ is a unit vector. □

REFERENCES

[Broi86] T. J. Broida and R. Chellappa, Estimation of object motion parameters from noisy images, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 90-99, 1986.

[Cram46] H. Cramér, *Mathematical Methods of Statistics*. Princeton Univ. Princeton, New Jersey, 1946.

[Gior85] A. A. Giordano and F. M. Hsu, *Least Squares Estimation with Applications to Digital Signal Processing*, Wiley, New York, 1985.

[Leve44] K. Levenberg, A method for the solution of certain nonlinear problems in least squares, *Quart. Appl. Math.*, 2, 1944, pp. 164-168.

[Luen69] D. G. Luenberger, *Optimization by Vector Space Methods*, Wiley, New York, 1969.

[Luen82] D. G. Luenberger, *Linear and nonlinear programming*, 2nd ed., Addison-Wesley, Massachusetts, 1982.

[Marq63] D. W. Marquardt, An algorithm for least squares estimation of nonlinear parameters, *SIAM J. Appl. Math.*, 11, 1963, pp. 431-441.

[Rao75] C. R. Rao, *Linear Statistical Inference and Its Applications*, 2nd Ed., Wiley, New York, 1973.

[Sore80] H. W. Sorenson, *Parameter Estimation: Principles and Problems*, Marcel Dekker, New York, 1980.

[Weng87] J. Weng, T. S. Huang, and N. Ahuja, Error analysis of motion parameter determination from image sequences, in *Proc. the First International Conference on Computer Vision*, London, England, June 8-11, 1987, pp. 703-707.

[Weng88a] J. Weng, N. Ahuja, and T. S. Huang, Two-view matching, in *Proc. 2nd International Conference on Computer Vision*, Florida, Dec. 1988, pp. 64-73.

[Weng88b] J. Weng, N. Ahuja, and T. S. Huang, Closed-form solution + maximum likelihood: a robust approach to motion and structure estimation, in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, Michigan, June 5-9, 1988, pp. 381-386.

[Weng89a] J. Weng, T. S. Huang and N. Ahuja, Motion from images: image matching, parameter estimation and intrinsic stability. in *Proc. IEEE Workshop on Visual Motion*, Irvine, CA, March 20-22, 1989, pp. 359-366.

[Weng89b] J. Weng, T. S. Huang, and N. Ahuja, Motion and structure from point correspondences: algorithm, error

analysis and error estimation, *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, No. 5, May 1989.

Table 1.a

Data and Results for Images of a Laboratory Scene (Mac Scene)			
Translation	0.016152	0.990948	0.133270
Rotation axis	0.966355	0.175804	-0.187752
Rotation angle		1.611214°	
Image error		0.000326	
Pixel width		0.000938	

Table 1.b

Estimated Errors for Images of the Mac Scene			
Errors of $\hat{\mathbf{T}}$	0.0026	0.0011	0.012
Errors of Rotation axis	0.0091	0.021	0.023
errors of rotation angle		0.14°	
Relative errors of $\hat{\mathbf{T}}$		0.012	
Relative errors of rotation axis		0.032	

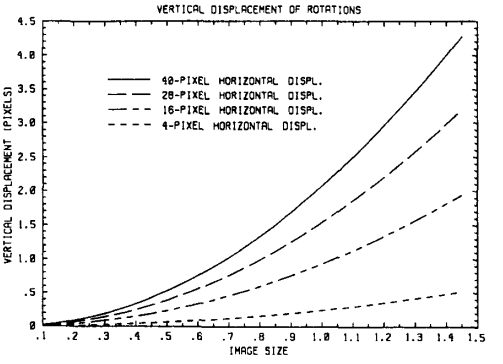

Fig. 4. Vertical displacement of a rotation about **X** axis versus image size for different horizontal displacement. Image point: center of a upper right quadrant of the image plane. Image resolution: 512×512.

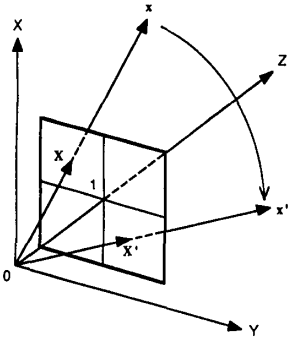

Fig. 1 Camera model and moving points.

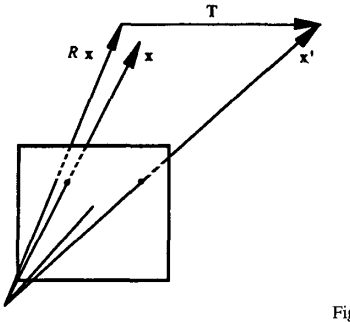

Fig. 2. Epipolar constraint: $R\mathbf{X}$, $\mathbf{T}$ and $\mathbf{X}'$ are coplanar

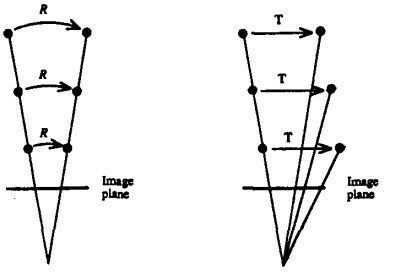

Fig. 5. Rotation and translation yield different displacement fields (a two-dimensional illustration). For a rotation, all the points on a projection line have the same projections after motion. For a translation, those points on a projection line have different projections after motion.

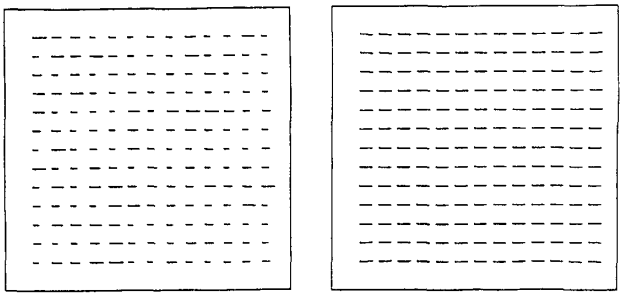

Fig. 3. Displacement field of a horizontal translation and a rotation about vertical axis. Resolution: 512×512

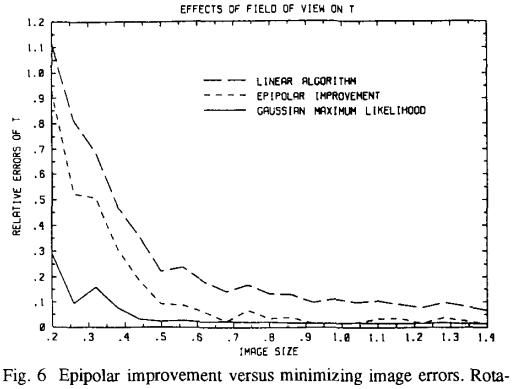

Fig. 6 Epipolar improvement versus minimizing image errors. Rotation axis: (1, 1, 1); Rotation angle: 1°. Translation: $(t_x, 0, 0)$, t_x varies linearly from 0.1 (for image size 0.2) to 0.7 (for image size 1.4). 100 random trials.

Table 2.a

Data and Results for Images of the Path Scene			
Translation	0.093230	-0.074830	0.992829
Rotation axis	0.443044	0.355395	0.823047
Rotation angle		0.145816°	
Image error		0.000223	
Pixel width		0.000938	

Table 2.b

Estimated Errors for Images of the Path Scene			
Errors of $\hat{\mathbf{T}}$	0.0023	0.0031	0.00031
Errors of Rotation axis	0.068	0.044	0.036
errors of rotation angle		0.012°	
Relative errors of $\hat{\mathbf{T}}$		0.0039	
Relative errors of rotation axis		0.088	

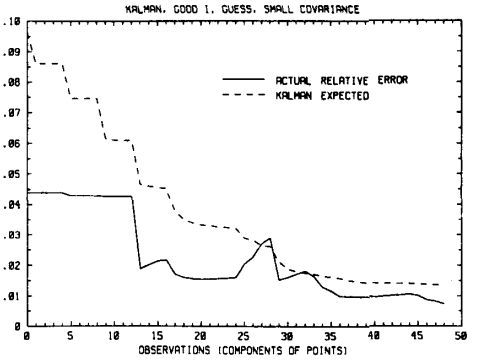

Fig. 7(a)

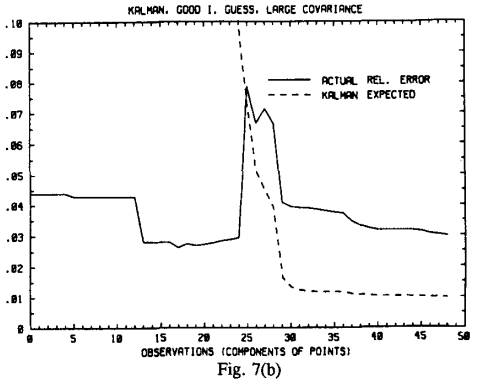

Fig. 7(b)

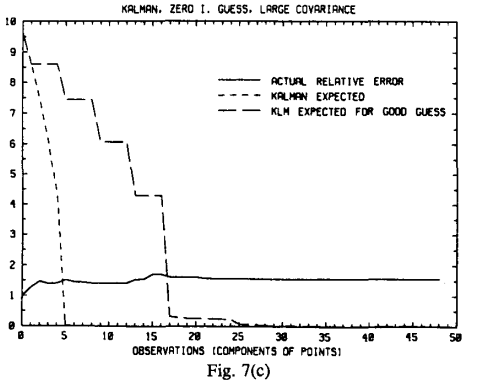

Fig. 7(c)

Fig. 7. Iterated extended Kalman least-squares filter for the nonlinear problem: a sample sequence. Rotation axis: (1, 1, 1); Rotation angle: 3°; Translation vector: (1.732, 1.732, −1.732). "Kalman expected" is the error predicted by the covariance matrix P_k. (a): Good initial guess generated by the linear algorithm with a small $P_{0,-1}$. (b): Same good initial guess as (a) with a large $P_{0,-1}$. The covariance of EIKF is not shown here for the cases with less than 25 observations (for a clear display of other values), and is shown in (c) instead under "IEKF expected for good i. guess". (c): Zero initial guess with a large $P_{0,-1}$.

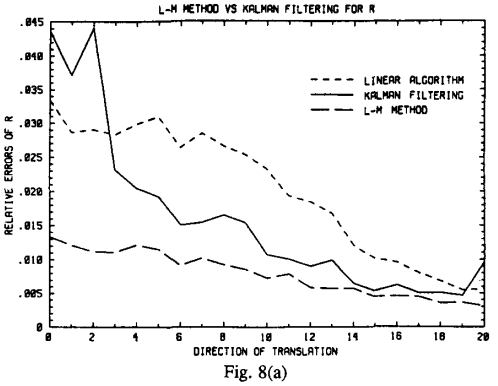

Fig. 8(a)

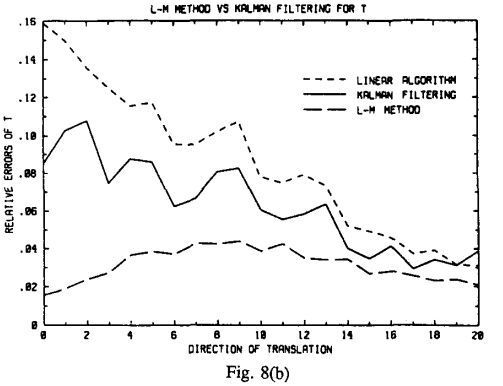

Fig. 8(b)

Fig. 8. Relative Errors of the linear algorithm, batch solution (Levenberg-Marquardt method) and sequential solution (IEKF). Rotation axis: (1, 1, 1); Rotation angle: 3°; For horizontal index from 0 to 20, the direction of translation changes from (1, 0, 0) to (0, 0, 1) in X-Z plane at evenly spaced 21 steps. The length of the translation vector is equal to 2.1 units. 100 random trials. (a): R; (b): $\mathbf{T}$; (c): Depths; (d): Image errors.

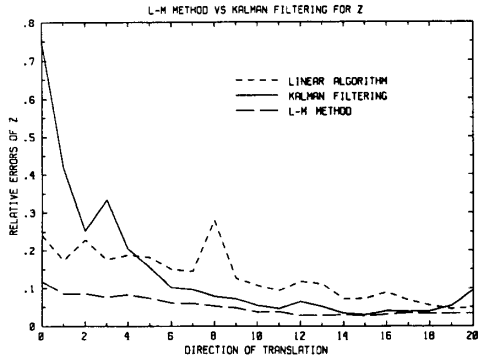

Fig. 8(c)

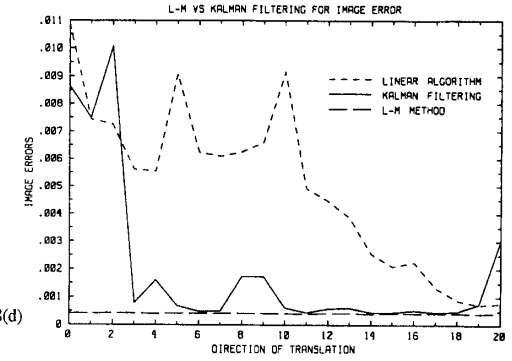

Fig. 8(d)

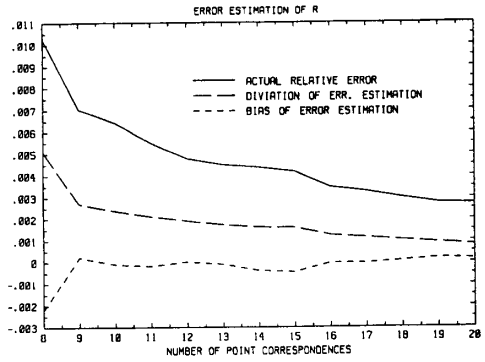

Fig. 9(a)

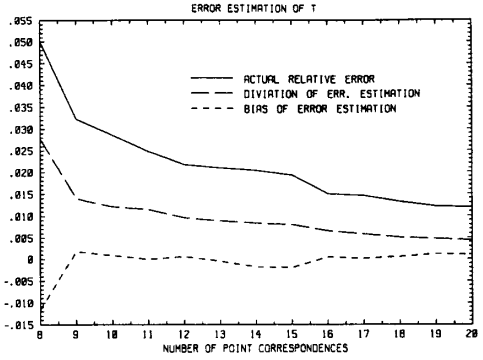

Fig. 9(b)

Fig. 9. Statistical record of error estimation. Actual relative error, deviation of error estimation and bias of error estimation for (a) R, (b) $\mathbf{T}$ vs. Number of Point Correspondences. Rotation axis: (1, 0.9, 0.8). Rotation angle: 5°. Translation: (0.5, −0.5, −3.0). 40 random trials.

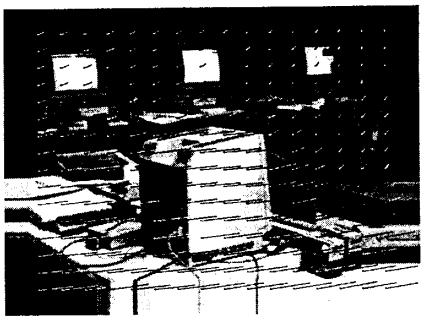

Fig. 10(a)

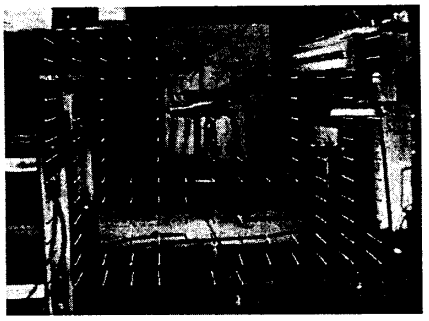

Fig. 11. Samples of the displacement field computed for the Path scene superimposed on the first image.

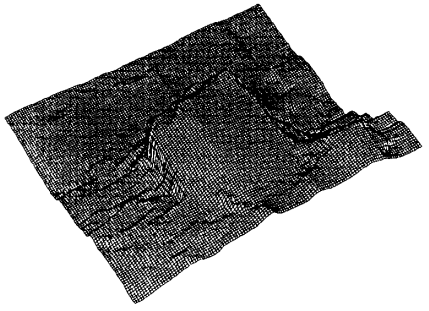

Fig. 10(b)

Fig. 10. (a) Samples of the displacement field computed superimposed on the extended first image of Mac scene. (b) The plot of $1/z$ (z is depth).