Optimal Motion and Structure Estimation

Juyang Weng, *Member, IEEE*, Narendra Ahuja, *Fellow, IEEE*, and Thomas S. Huang, *Fellow, IEEE*

***Abstract*— The existing linear algorithms exhibit various high sensitivities to noise. The analysis presented in this paper provides insight into the causes for such high sensitivities. It is shown in this paper that even a small pixel-level perturbation may override the epipolar information that is essential for the linear algorithms to distinguish different motions. This analysis indicates the need for optimal estimation in the presence of noise. Then, we introduce methods for optimal motion and structure estimation under two situations of noise distribution: 1) known and 2) unknown. Computationally, the optimal estimation amounts to minimizing a nonlinear function. For the correct convergence of this nonlinear minimization, we use a two-step approach. The first step is using a linear algorithm to give a preliminary estimate for the parameters. The second step is minimizing the optimal objective function starting from that preliminary estimate as an initial guess. A remarkable accuracy improvement has been achieved by this two-step approach over using the linear algorithm alone. In order to assess the accuracy of the optimal solution, the error in the solution of the optimal estimation algorithm is compared with a theoretical lower error bound—Cramér-Rao bound. The simulations have shown that with Gaussian noise added to the coordinates of the image points, the actual error in the optimal solution is very close to the bound. In addition, we also use the Cramér-Rao bound to indicate the inherent instability of motion estimation from small image disparities, such as motion from optical flow. Finally, it is known that given the same nonlinear objective function and the same initial guess, different minimization methods may lead to different solutions. We investigate the performance difference between a batch least-squares technique (Levenberg-Marquardt) and a sequential least-squares technique (iterated extended Kalman filter) for this motion estimation problem, and the simulations showed that the former gives better results.**

***Index Terms*— Cramér-Rao bound, extended Kalman filter, maximum likelihood estimation, minimum variance estimation, motion estimation, nonlinear least-squares, structure from motion.**

I. Introduction

TWO TYPES OF methods have been used for 3-D motion and structure analysis. The first type is iteratively solving nonlinear equations, which can be traced back to 1979 [39] when nonlinear equations were derived to relate 3-D motion parameters with the observables in the image plane. The challenge for these type of methods [39], [7], [57] is to solve these nonlinear equations. Although numerical methods could be applied to these nonlinear equations, the solution is not guaranteed, and as reported by a number of researchers, one may end up with a false solution if the initial guess is not sufficiently near the true value. The second type is solving the problem using linear algorithms [23], [46]. However, it has been reported that the solutions are highly sensitive to noise. This situation has raised concerns over whether the structure from motion problem itself is unstable.

In fact, a stable solution is possible if appropriate optimality conditions are enforced. The optimization approach presented here started in early 1986 [51], [52], and this is its journal version. Our approach to optimization was motivated by the following observations on linear algorithms.

1) For certain types of motion, even pixel-level perturbations (such as spatial digitization noise of conventional CCD video cameras) may override the information characterized by the epipolar constraint, which is a key constraint used for determining motion and structure by linear algorithms. The epipolar constraint only constrains one of the two components of the image position of a point.
2) Existing linear algorithms give closed-form solutions to motion parameters; however, the constraints in the intermediate parameter matrix are not fully used. It is useful to examine the constraint in the intermediate parameter matrix and use this constraint to improve the accuracy of the solution in the presence of noise.

The above considerations are unified under a general framework of optimal estimation: Given the noise-contaminated points, we want the best estimator for motion and structure parameters. The following are the highlights of the paper:

1) This paper investigates approaches to optimal estimation with known or unknown noise distributions.
2) Further, this paper introduces an approach to assessing the accuracy of the optimal solutions, which requires a method that is different from that for the linear algorithm [55].
3) Given an algorithm that computes a solution from noise-contaminated data, a fundamental question to ask is: Can one design an algorithm that gives solutions with higher accuracy? The questions of this type address the inherent stability issue of motion estimation. In this paper, we formulate the theoretical performance bounds for this problem and compare them with the actual performance. This study also enables us to quantitatively assess the inherent stability problem of estimating motion from small image disparities such as motion from optical flow.
4) The type of algorithms used for nonlinear optimization is crucial in determining whether the optimal solution can be reliably obtained. The sequential processing method (Kalman filtering) has been used in many applications

Manuscript received November 1, 1991; revised August 26, 1992. This work was supported by the National Science Foundation under grants ECS-83-52 408 and IRI-86-05 400. Recommended for acceptance by Associate Editor A. Blake.

The authors are with the Beckman Institute, University of Illinois, Urbana, Illinois 61801.

IEEE Log Number 9211194.

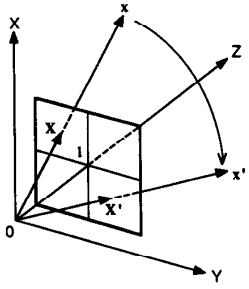


Fig. 1. Normalized camera model and a moving scene.

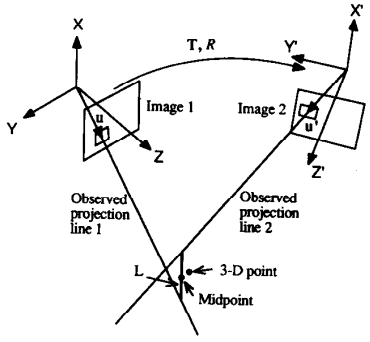


Fig. 2. Another view of the relative motion. Moving camera takes two images.

[42], [5], [10], [38], [11], [4], [31]. However, the problems with Kalman filtering have been largely neglected in this field. This paper analyzes sequential and batch processing algorithms and their performance differences for our motion problem. In fact, the performance differences between these two types of methods is quite large for this nonlinear problem.

II. Linear Algorithms and Their Stability

The problem of estimating motion and structure from point correspondences through two views can be formulated as follows. Two images are taken at different positions and orientations in a rigid environment. The objective is to estimate the relative motion between the camera and the environment as well as the structure of the visible scene.

Let the coordinate system be fixed on the camera as shown in Fig. 1. The image vector of the point $\boldsymbol{x} = (x, y, z)^t$ is defined by

$$\boldsymbol{X} = (u, v, 1)^t = (x/z, y/z, 1)^t$$

in the 3-D coordinate system. The image plane vector $\boldsymbol{u} = (u, v)^t = (x/z, y/z)^t$ is the projection of the 3-D point. Let R be the rotation matrix and $\boldsymbol{T}$ be the translation vector, and let $\boldsymbol{x}$ move to $\boldsymbol{x}'$ under the motion, that is

$$\boldsymbol{x}' = R\boldsymbol{x} + \boldsymbol{T}. \tag{2.1}$$

Similarly, define the image plane vector $\boldsymbol{u}'$ of the image vector $\boldsymbol{X}'$. $\boldsymbol{X}' = (u', v', 1)^t = (x'/z', y'/z', 1)^t$.

Equivalently, the relative motion can also be viewed as that due to the motion of the camera. Let the world coordinate system be fixed with the scene and coincide with the camera coordinate system at time t_1 as shown in Fig. 2. To result in the same images as before, the motion of the camera can be represented by a "reverse motion," that is, a translation $-\boldsymbol{T}$ followed by a rotation R^t in the world coordinate system (namely, any point $\boldsymbol{p}$ on the camera is moved to $\boldsymbol{p}'$ in the world coordinate system), and $\boldsymbol{p}$ and $\boldsymbol{p}'$ are related by

$$\boldsymbol{p}' = R^t(\boldsymbol{p} - \boldsymbol{T}) \tag{2.2}$$

in the world coordinate system (see Fig. 2). We will mainly consider the case where the camera is stationary. When it is necessary, we will give the interpretations for the case of a moving camera.

A. Linear Algorithms

Different versions of linear algorithms have been published in literature (a partial list would include [23], [46] [59], [11], and [49]). Although those algorithms use different ways to determine the unknowns, they share the same key structure: determining intermediate parameters, which are called essential parameters, based on the epipolar constraint. To be specific, we use the algorithm in [55] as an example.

B. The Epipolar Constraint

The key constraint that the linear algorithms employ to solve for motion parameters is that $\boldsymbol{X}'$, $R^t\boldsymbol{X}$, and $\boldsymbol{T}_s$ must be linearly dependent (or coplanar) according to (2.1), or equivalently, the vector triple product vanishes:

$$(\boldsymbol{X}')^t(\boldsymbol{T}_s \times (R\boldsymbol{X})) = 0 \tag{2.3}$$

where $\times$ denotes vector cross product. Its geometrical illustration is shown in Fig. 3. We define a mapping $[\cdot]_\times$ from a 3-D vector to a 3×3 matrix:

$$\left[(x_1, x_2, x_3)^t\right]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}.$$

Using this mapping, we can express the cross product of two vectors by the matrix multiplication of a 3×3 matrix and a column matrix: $\boldsymbol{X} \times \boldsymbol{Y} = [\boldsymbol{X}]_\times \boldsymbol{Y}$. Define the essential matrix E to be

$$E = [\boldsymbol{T}_s]_\times R \tag{2.4}$$

where $\boldsymbol{T}_s$ is a unit vector such that $\boldsymbol{T} \times \boldsymbol{T}_s = \boldsymbol{0}$. Equation (2.3) can then be rewritten as

$$(\boldsymbol{X}')^t E \boldsymbol{X} = 0. \tag{2.5}$$

Equation (2.5) is linear in the elements of matrix E. Using eight or more point correspondences, the linear algorithms first solve for E based on (2.5) and then solve for motion parameters from E.

The plane in which $\boldsymbol{X}'$, $\boldsymbol{T}$, and $R\boldsymbol{X}$ lie is called the epipolar plane of the point. Its intersection with the image plane is called the epipolar line of the point. The constraint that $R\boldsymbol{X}$, $\boldsymbol{T}$, and $\boldsymbol{X}'$ are coplanar is called the epipolar constraint.

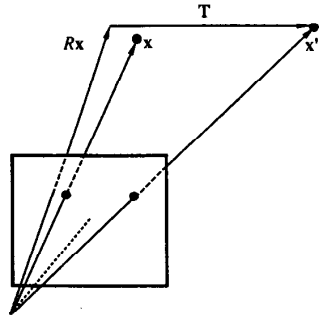


Fig. 3. Epipolar line constraint: $R\boldsymbol{X}$, $\boldsymbol{T}$, and $\boldsymbol{X}'$ are coplanar.

We will show below that only one component of the image position of a point is used by the epipolar constraint. In fact, consider two unit vectors in the image plane from a point $\boldsymbol{X}'$: One is aligned with the epipolar line (denoted by $\boldsymbol{\delta}_a$), and the other is perpendicular to the epipolar line (denoted by $\boldsymbol{\delta}_p$). Any perturbed image position of $\boldsymbol{X}'$ can be represented by $\boldsymbol{X}'(\epsilon) = \boldsymbol{X}' + a\boldsymbol{\delta}_a + b\boldsymbol{\delta}_p$ for some real numbers a and b. Since $\boldsymbol{\delta}_a^t(\boldsymbol{T}_s \times R\boldsymbol{X}) = 0$, from (2.3), we have $\boldsymbol{X}'(\epsilon)^t(\boldsymbol{T}_s \times R\boldsymbol{X}) = \boldsymbol{\delta}_p^t(\boldsymbol{T} \times R\boldsymbol{X})$. In other words, the perturbation of $\boldsymbol{X}'$ along the epipolar line direction does not affect the value of $\boldsymbol{X}'(\epsilon)^t(\boldsymbol{T}_s \times R\boldsymbol{X})$. The location of the points on the epipolar line is irrelevant to the epipolar constraint. It is related to the depth of the point as well as the motion parameters.

It has been proved that based on the epipolar constraint, the rotation and translation parameters can be solved uniquely from image vectors of a nondegenerate configuration of 3-D points. However, the questions to ask include the following:

1) The essential matrix E has only five degrees of freedom (two for unit vector $\boldsymbol{T}_s$ and three for rotation matrix R). How can the constraint in E be used to improve the accuracy in the presence of noise?
2) How reliably the motion parameters can be estimated using just the epipolar constraint?
3) Can another component (along the epipolar line) of the image points be used, in addition to the epipolar constraint, to improve the reliability of the estimated motion and structure parameters in the presence of noise?

These problems are investigated in the following sections.

C. Using the Constraint in the Essential Matrix

By definition of (2.4), E has only five degrees of freedom. E should be the product of a skew symmetric matrix ($S = -S^t$) and a rotation matrix R (orthonormal with determinant 1).

Theorem: Given a 3×3 matrix E, the necessary and sufficient condition for an existing rotation matrix R and a unit vector $\boldsymbol{T}_s$, such that $E = [\boldsymbol{T}_s]_\times R$, is that the eigenvalues of E^tE are $0, 1, 1$, respectively.

Proof: See [18].

The constraint on the eigenvalues of E^tE can be written as polynomial equations in terms of elements of E. However, polynomial equations introduce spurious solutions. In the presence of noise, E estimated from the linear equations generally does not satisfy the conditions in the Theorem. This causes errors in the solutions of R and $\boldsymbol{T}_s$.

The constraint in E can be used by iteratively improving the computed R and $\boldsymbol{T}_s$ to minimize the weighted sum of $\left((\boldsymbol{X}')^t(\boldsymbol{T}_s \times R\boldsymbol{X})\right)^2$. The weight is the reciprocal of the error variance of $(\boldsymbol{X}')^t(\boldsymbol{T}_s \times R\boldsymbol{X})$. Assuming the components of $\boldsymbol{u}$ and $\boldsymbol{u}'$ have additive uncorrelated zero mean noise with variance σ^2, it is shown in Appendix A that the variance of the first-order error of $(\boldsymbol{X}')^t(\boldsymbol{T}_s \times R\boldsymbol{X})$ is given by

$$\sigma^2\left(||R^t(\boldsymbol{T}_s \times \boldsymbol{X}')||_{z=0}^2 + ||\boldsymbol{T}_s \times R\boldsymbol{X}||_{z=0}^2\right) \tag{2.6}$$

where $||(a, b, c)||_{z=0}^2 \triangleq a^2 + b^2$.

Some comments are in order here:

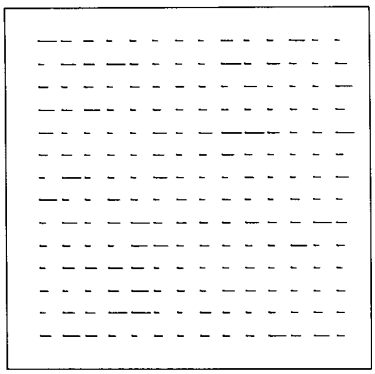
1) Since the weights include the unknowns R and $\boldsymbol{T}_s$, one cannot use those weights in solving (2.5) without using iterations.
2) To ensure that the equations to be solved are all linear, the constraints in E stated in the above theorem cannot be used either since those constraints are nonlinear.
3) If those constraints are used together with (2.5), generally fewer than eight points are need to solve for motion parameters. However, this again requires solving nonlinear equations.
4) More constraints beside the epipolar constraint can be used in the presence of noise. The epipolar constraint alone cannot always ensure a reliable solution, which we will discuss in the next section.

D. A Type of Motion

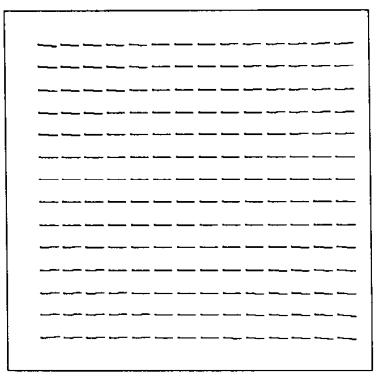
The reliability of the estimated motion and structure parameters depends on many factors, including structure of the scene, motion parameters, and imaging system parameters. The effects of those factors on the reliability of the estimates are discussed qualitatively in [55]. Here, we quantitatively analyze the fact that even small errors can override the information used by the epipolar constraint.

Let us consider two corresponding types of motion: One is a pure translation, and the other is a pure rotation. For the type of translation, the translation vector and the optical axis are orthogonal. For the type of pure rotation, the rotation axis is orthogonal to both the optical axis and the translation vector of the pure translation. Without loss of generality, let the translation direction be aligned with the y axis and the rotation axis be aligned with the x axis. Fig. 4 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 a horizontal pure translation $\boldsymbol{T} = (0, t_2, 0)^t$, $t_2 \neq 0$, from $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{T}$, we have

$$\begin{aligned} u' &= x'/z' = x/z = u, \\ v' &= y'/z' = (y + t_2)/z = v + t_2/z \neq v. \end{aligned} \tag{2.7}$$



(a)



(b)

Fig. 4. Two image plane displacement fields: (a) From a horizontal translation; (b) rotation about a vertical axis. The field from the translation is exactly horizontal. The field from the rotation is almost horizontal since the maximum vertical displacement is just 2.9 pixels. Image size: 0.7. resolution: 512×512.

The displacement vector on image plane is equal to

$$\boldsymbol{u}' - \boldsymbol{u} = (u' - u, v' - v)^t = (0, v' - v)^t.$$

The left-hand side of (2.3) for this case is

$$(\boldsymbol{X}')^t(\boldsymbol{T}_s \times R\boldsymbol{X}) = u' - u$$

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

For the pure rotation, the rotation matrix is given by

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

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

$$u' - u = \frac{u}{v\sin\theta + \cos\theta} - u = \frac{u(1 - \cos\theta - v\sin\theta)}{v\sin\theta + \cos\theta}, \quad (2.8)$$

$$v' - v = \frac{v\cos\theta - \sin\theta}{v\sin\theta + \cos\theta} - v = -\frac{(1 + v^2)\sin\theta}{v\sin\theta + \cos\theta}. \quad (2.9)$$

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 very close to zero: Assume

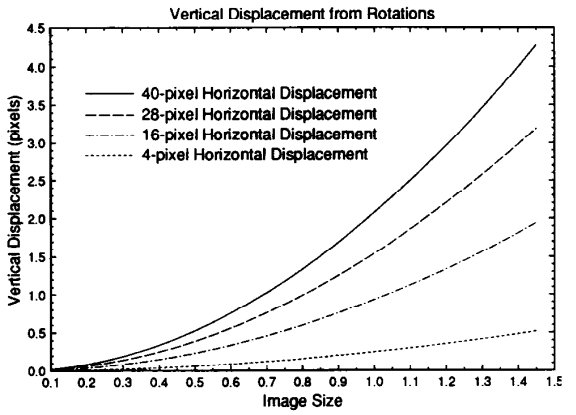


Fig. 5. Vertical displacement of a rotation about the x axis versus image size for different horizontal displacement. Image point: center of an upper right quadrant of the image plane. Image resolution: 512×512.

the image size is s (image is a $s \times s$ square) with $m \times m$ resolution (pixels). The pixel size is then s/m. The vertical displacement in (2.8) in terms of the number of pixels is then

$$\frac{(u' - u)m}{s} = \frac{u(1 - \cos\theta - v\sin\theta)m}{s(v\sin\theta + \cos\theta)}. \quad (2.10)$$

Fig. 5 shows the value of the vertical displacement based on (2.10), in terms of number of pixels, at the center of a quadrant of an image with 512×512 resolution, for different horizontal displacement $v' - v$, and different image sizes s. For example, for image size 1 with a unit focal length (roughly equivalent to a 35-mm wide-angle lens of a 35-mm camera), the vertical displacement is just about 2 pixels for a large 40-pixel horizontal displacement. For small motions with 4-pixel image displacement (similar or even smaller motion is generally required for optical flow approaches), the vertical displacement is less than a half pixel even with a very wide field of view. This implies that in the presence of small errors in the image coordinates (e.g., in a magnitude of one or two pixels), pure translation can be interpreted by a rotation as far as the epipolar constraint is concerned, and vice versa. In other words, the epipolar constraint cannot disambiguate the translation from the corresponding rotation in the presence of even small image digitization noise.

Now, let us consider the factor of decreasing the field of view or, equivalently, the image size with unit focal length. From Fig. 5, we can see that the vertical displacement, in terms of number of pixels, approaches zero as the image size decreases with a constant image resolution and a constant image horizontal displacement (in pixels). In fact, the vertical displacement, in terms of pixels, decreases quadratically as the image size decreases. This fact can be derived from (2.10). For a small rotation with angle θ, the horizontal displacement is in the same order as θ (remember that we have a unit focal length). To fix the amount of horizontal displacement in the image plane with respect to image size s, let $\theta = sk$, where k is a constant. For a point (u, v) fixed relatively in the image frame, $|u| = sk_u$ and $|v| = sk_v$, where k_u and k_v are

constants. Letting s go to zero, the absolute value of (2.10) is in the order of

$$\frac{mu}{s}\left(\frac{\theta^2}{2} - v\theta\right) = mkk_u s^2\left(\frac{k}{2} - k_v\right) \tag{2.11}$$

which is quadratic in s. We have showed that the number of pixels of vertical displacement goes to zero quadratically as the image size s approaches zero. Therefore, the vertical displacement decreases much faster than the pixel size. This result implies that motion estimation based on only the epipolar constraint is inherently very unreliable with a small field of view since the epipolar constraint relies on the amount of vertical displacement to disambiguate a pure rotation from the pure translation.

In summary, we have shown that for each lateral translation (parallel to the image plane), there exists a corresponding type of rotation such that the displacement field of translation can be interpreted by the rotation without significantly violating the epipolar constraint. In the presence of even small pixel-level noise, the displacement of the translation can be interpreted by appropriate rotation and vise versa. Therefore, small pixel-level errors will cause large errors in the estimated R and $\boldsymbol{T}_s$. Worse still, once a lateral translation is mistakenly interpreted as the corresponding pure rotation, the estimated translation direction can be arbitrary since pure rotation is an inherently degenerate case. Up until now, we have answered the second question raised at the end of Section II-B: There is a large class of motion with which the motion parameters cannot be estimated reliably using just the epipolar constraint.

There are two ways to improve the estimated motion parameters. The first one is to use a large number of points (or a dense displacement field). If the measurement error has a zero mean and is highly random, the solution error tends to be overcome in the solution of an overdetermined system. This is true for the cases of our computer simulations where the noise is generated by a zero mean pseudo-random number generator. However, with the displacement field (or point correspondences) computed automatically by an algorithm, the measurement errors are often biased, and the amount of bias is usually unknown. This fact makes the overdetermination less effective.

The second way is to use both components of the features in the image plane, which gives an answer to the third question raised at the end of Section II-B and will be discussed in the following subsection.

E. Beyond the Epipolar Constraint

From (2.3), one can see that the depths of the object points are excluded from the epipolar constraint. This is desirable to the linear algorithms since the depths of the points are unknown. The epipolar constraint uses only one component of the coordinates of the image points: $\boldsymbol{X}'$ can be any image vector in the epipolar plane where $\boldsymbol{T}_s$ and $R\boldsymbol{X}$ lie. However, the other component left out by the epipolar constraint is important for determining motion parameters due to the following properties:

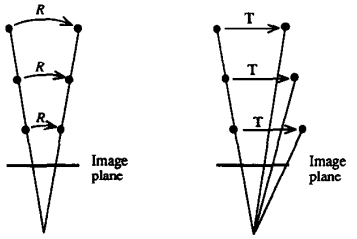


Fig. 6. Rotation and translation yield different displacement fields (a 2-D illustration). For a rotation, all the points on a projection line have the same projections after motion. For a translation, those points on a projection line have different projections after motion.

1) Under a rotation, all the points on a projection line (passing through the origin and an image point) project on to the same image point after rotation.
2) Under a translation, those points on a projection line have different projections after translation: the closer the point is to the image plane, the larger the displacement in the image plane.

Fig. 6 illustrates these properties. The first property is obvious. For the second property, let $\boldsymbol{T} = (t_x, t_y, t_z)^t$. From $\boldsymbol{x}' = \boldsymbol{x} + \boldsymbol{T}$, after some algebraic manipulations, we have

$$||\boldsymbol{u}' - \boldsymbol{u}||^2 = \left((t_x - t_z u)^2 + (t_y - t_z v)^2\right)/(z + t_z)^2. \tag{2.12}$$

Notice that $z + t_z = z' > 0$. With $\boldsymbol{u}$ and $\boldsymbol{T}$ fixed, the larger the z, the smaller the magnitude of the image displacement.

From (2.12), we know that the magnitude of the image displacement is inversely proportional to the depth z'. As long as $1/z' = 1/(z + t_z)$ has a large variation among points, the displacement field is quite different between a translation and a rotation. These differences are useful for distinguishing a rotation from a translation. For example, in Fig. 4, the image displacement vectors are almost horizontal for both rotation and translation. However, the lengths of the displacement vectors are quite different with the translation but are similar with the rotation. It is impossible to interpret a translation by a rotation if both components of the points are used.

With the aim of a closed-form solution, the linear algorithms have to exclude information that is related to point depths. Only the component of the points that is independent with depth is used by the epipolar constraint. Mathematically, this is enough to determine motion parameters. However, in the presence of noise, disregarding information that is related to the structure of the scene results in less reliable estimates of motion parameters since this information is also related to motion (Figs. 4 and 6). The optimization approach we will introduce in the following section makes use of both components of the image points in an integrated way. As shown by simulations, this significantly improves accuracy over using the epipolar constraint only, especially with a relatively small field of view where the epipolar constraint is particularly week in determining motion (see Fig. 5). The next two sections discuss these optimization methods.

III. Optimal Motion Estimation with Known Noise Distribution

Before discussing optimality, we briefly review some objective functions other algorithms have used. From optical flow, Bruss and Horn [7] propose an approach to minimizing some measure of the discrepancy between the measured flow and that predicted from the computed motion parameters. Mitiche and Aggarwal [30] employ a rigidity criterion: Using depth of each point as parameters, the sum of squared changes in the point-to-point distances, before and after motion, is to be minimized. The linear algorithms in [11] and [49] employ least-squares criteria for different equations.

A more statistically sound way of estimating the parameters is using the information about error (or noise) distribution. In reality, the feature locations and their displacement vectors in the image plane are the results from a feature detector and the corresponding matcher, whose accuracy is influenced by a variety of factors including the condition of lighting, the structure of the scene, the accuracy of the system calibration, image resolution and the performance of the feature detecting, and matching algorithms. The observed 2-D image plane vectors $\boldsymbol{u}_i$ of image 1 and $\boldsymbol{u}_i'$ of image 2 are noise-contaminated versions of the true ones. In other words, $(\boldsymbol{u}_i, \boldsymbol{u}_i')$ is the observed value of a pair of random vectors $(\boldsymbol{U}_i, \boldsymbol{U}_i')$. What we obtain is a sequence of the observed image vector pairs

$$\boldsymbol{u} \triangleq \left(\boldsymbol{u}_1^t, (\boldsymbol{u}_1')^t, \boldsymbol{u}_2^t, (\boldsymbol{u}_2')^t, \cdots, \boldsymbol{u}_n^t, (\boldsymbol{u}_n')^t\right)^t$$

of a sequence of random vector pairs

$$\boldsymbol{U} \triangleq \left(\boldsymbol{U}_1^t, (\boldsymbol{U}_1')^t, \boldsymbol{U}_2^t, (\boldsymbol{U}_2')^t, \cdots, \boldsymbol{U}_n^t, (\boldsymbol{U}_n')^t\right)^t.$$

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

$$\boldsymbol{X} \triangleq \left(\boldsymbol{x}_1^t, (\boldsymbol{x}_1')^t, \boldsymbol{x}_2^t, (\boldsymbol{x}_2')^t, \cdots, \boldsymbol{x}_n^t, (\boldsymbol{x}_n')^t\right)^t.$$

Let the probability density function of $\boldsymbol{U}$, given $\boldsymbol{M} = \boldsymbol{m}$ and $\boldsymbol{X} = \boldsymbol{x}$, be $p_{\boldsymbol{U}|\boldsymbol{M},\boldsymbol{X}}(\boldsymbol{u}|\boldsymbol{m}, \boldsymbol{x})$. The maximum likelihood estimates of motion parameters $\boldsymbol{m}^*$ and scene structure $\boldsymbol{x}^*$ are such that the density $p_{\boldsymbol{U}|\boldsymbol{M},\boldsymbol{X}}(\boldsymbol{u}|\boldsymbol{m}, \boldsymbol{x})$ reaches the maximum, namely

$$p_{\boldsymbol{U}|\boldsymbol{M},\boldsymbol{X}}(\boldsymbol{u}|\boldsymbol{m}^*, \boldsymbol{x}^*) \geq p_{\boldsymbol{U}|\boldsymbol{M},\boldsymbol{X}}(\boldsymbol{u}|\boldsymbol{m}, \boldsymbol{x}) \tag{3.1}$$

holds for all possible motion parameters $\boldsymbol{m}$ and scene structure $\boldsymbol{x}$. The motivation for using maximum likelihood criterion is that, among others, under fairly general conditions, the estimator is consistent (it converges in probability to the correct value when the number of observations approaches infinity), asymptotically unbiased, asymptotically Gaussian, and asymptotically efficient (as the number of observations approaches infinity, it is unbiased, has finite covariance, and there is no other unbiased estimate whose covariance is smaller) [8], [45], [56], [28].

A. Gaussian Noise in Image Plane

Gaussian distribution is commonly used for modeling noise. Intuitively, if the errors arise from many sources and are influenced by the sum of many factors, the distribution is roughly Gaussian by the central limit theorem.

Now, assume that each image coordinate of the observed projection has an additive zero mean Gaussian noise. Therefore, we assume that the distributions, given motion and structure, are independent between different points. When the distance between two points is not very small, compared with pixel size, such an assumption of independence is reasonable. Let $\boldsymbol{h}_i(\boldsymbol{m}, \boldsymbol{x})$ be the noise-free projection of the ith point in the first image, given motion $\boldsymbol{m}$ and structure $\boldsymbol{x}$, and let $\boldsymbol{h}_i'(\boldsymbol{m}, \boldsymbol{x})$ be the corresponding projection in the second image. Then, after some simple manipulation, we know that the maximum likelihood estimator is the one that minimizes

$$\sum_{i=1}^{n} \left(||\boldsymbol{u}_i - \boldsymbol{h}_i(\boldsymbol{m}, \boldsymbol{x})||^2 + ||\boldsymbol{u}_i' - \boldsymbol{h}_i'(\boldsymbol{m}, \boldsymbol{x})||^2\right) \tag{3.2}$$

which is simply the sum of discrepancies between the observed and the inferred projections. We define the *image plane error*, or simply *image error*, as

$$\left[\frac{1}{2n}\sum_{i=1}^{n} \left(||\boldsymbol{u}_i - \boldsymbol{h}_i(\boldsymbol{m}, \boldsymbol{x})||^2 + ||\boldsymbol{u}_i' - \boldsymbol{h}_i'(\boldsymbol{m}, \boldsymbol{x})||^2\right)\right]^{1/2}. \tag{3.3}$$

The image error can be easily extended to a weighted version when we have knowledge about the reliability of each point.

Although the noise in the 2-D image is Gaussian, the 3-D distribution of the error in the 3-D points is obviously no longer Gaussian, as shown in Fig. 7. This shape of 3-D uncertainty is easily taken into account by the minimization of the 2-D image plane error.

B. Noise with a Limited Extent

Let us briefly consider another case where the noise magnitude is confined to a small range. This example is useful later in our experiments to indicate the impact of variation in noise distribution.

With digitization noise, for example, the true projection of a point is confined to a rectangle centered at the observed image position of the point. Such a rectangle is called the uncertainty rectangle. The true 3-D position of the point is confined to an infinite pyramid defined by the focal point as the apex and the uncertainty rectangle as a cross section, as shown in Fig. 8. The second view defines another pyramid for the point. The 3-D point must lie inside the volume of intersection of these two pyramids. Such an intersection of the pyramids is called the *uncertainty polyhedron*.

When the observed image positions of a point, which are determined by motion parameter vector $\boldsymbol{m}$ and structure $\boldsymbol{x}$, are exactly correct, the corresponding uncertainty polyhedron has a volume $V(\boldsymbol{m}, \boldsymbol{x})$. If those image positions are perturbed by noise, the volume of the uncertainty polyhedron will generally decrease to a value $v(\boldsymbol{u}, \boldsymbol{m}, \boldsymbol{x})$, where $\boldsymbol{u}$ is a noisy observation vector. Therefore, we assume that the probability for a point

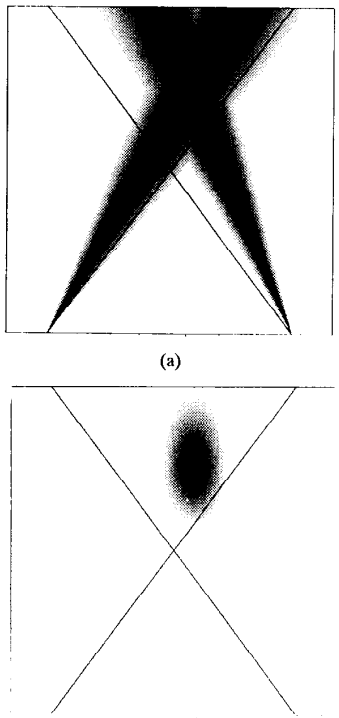


Fig. 7. Two-dimensional illustration of point distribution in 3-D based on Gaussian error distribution in the image planes. Darker areas have higher probability. Two diagonal lines across the figure are optical axes of two cameras: (a) Two observed projection lines, with error distribution, intersect; (b) two observed projection lines, with error distribution, determine the distribution of the point in 3-D.

to be confined to the uncertainty polyhedron with a volume v is equal to v/V.

For simplicity of computation, we assume that the probability of a point lying in the intersection is independent from point to point. For the observed point i, the volumes v and V discussed above are denoted by v_i and V_i, respectively. The event observed is that the feature points are all confined to the corresponding uncertainty polyhedrons. Thus, the probability that all points lie in the corresponding intersection can be written as

$$\prod_{i=1}^{n} \frac{v_i(\boldsymbol{u}, m, x)}{V_i(\boldsymbol{m}, x)}. \tag{3.4}$$

We call this probability model *the uncertainty polyhedron model*. Obviously, this probability does not characterize the actual probability exactly since the actual "density" within each uncertainty polyhedron is not uniform. Since the size of each uncertainty polyhedron is actually very small compared with the object to camera distance, such a nonuniformity can be neglected without degrading performance significantly. The maximization of the objective function in (3.4) has been implemented by a numerical method [52], and the details are omitted here.

An alternative way to consider noise distribution with a limited extent is modeling noise directly in the 2-D image plane, but this will lead to a complex objective function that has little practical value.

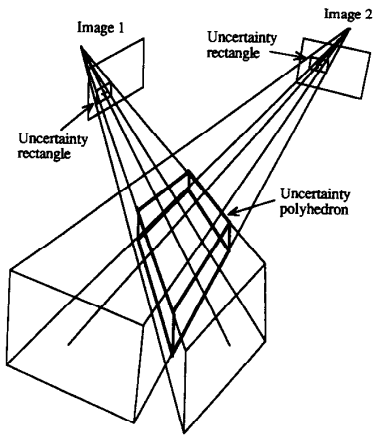


Fig. 8 Intersection of two pyramids defines uncertainty polyhedron.

IV. Optimal Motion Estimation with Unknown Noise Distribution

The computation of the maximum likelihood estimate requires the knowledge of actual noise distribution and the solution of likelihood equation. This causes difficulties in practice. First, the distribution of noise is usually unknown. Second, even if the noise distribution is known, the solution of the maximum likelihood estimator is often very difficult to obtain analytically. In this section, we discuss an optimal estimator for general error distributions.

A. Minimum Variance

Let $\boldsymbol{m}$ be the parameter vector to be estimated and $\hat{\boldsymbol{m}}(\boldsymbol{u})$ be the estimator based on the observation vector $\boldsymbol{u}$. The error vector of $\hat{\boldsymbol{m}}$ is $\boldsymbol{\delta_m} \triangleq \hat{\boldsymbol{m}} - \boldsymbol{m}$. The estimator $\hat{\boldsymbol{m}}$ that minimizes

$$\mathbf{E}||\boldsymbol{\delta_m}||^2 = \mathbf{E}||\hat{\boldsymbol{m}} - \boldsymbol{m}||^2 \tag{4.1}$$

is called the minimum variance estimator (or least-squares estimator or minimum mean square estimator). Let us first consider a linear problem.

Suppose

$$\boldsymbol{y} = A\boldsymbol{m} + \boldsymbol{\delta_y} \tag{4.2}$$

where $\boldsymbol{\delta_y}$ is a random vector with zero mean $\mathbf{E}\boldsymbol{\delta_y} = \mathbf{o}$ and covariance matrix $\Gamma_{\boldsymbol{y}} = \mathbf{E}\boldsymbol{\delta_y}\boldsymbol{\delta_y}^t$. According to Gauss-Markov theorem [24], [41], [14], the unbiased linear minimum variance estimator of $\boldsymbol{m}$ is

$$\hat{\boldsymbol{m}} = \left(A^t\Gamma_{\boldsymbol{y}}^{-1}A\right)^{-1}A^t\Gamma_{\boldsymbol{y}}^{-1}\boldsymbol{y} \tag{4.3}$$

whose error covariance matrix is

$$\Gamma_{\hat{\boldsymbol{m}}} \triangleq \mathbf{E}(\hat{\boldsymbol{m}} - \boldsymbol{m})(\hat{\boldsymbol{m}} - \boldsymbol{m})^t = \left(A^t\Gamma_{\boldsymbol{y}}^{-1}A\right)^{-1}.$$

In other words, among all the possible estimators of the form $\hat{\boldsymbol{m}} = L\boldsymbol{y}$, where L is any matrix, the estimator in (4.3) minimizes $\mathbf{E}||\hat{\boldsymbol{m}} - \boldsymbol{m}||^2$.

The estimator in (4.3) is equivalent to the least-squares estimator with weight matrix $\Gamma_{\boldsymbol{y}}^{-1}$, i.e., the estimator minimizes

$$(\boldsymbol{y} - A\boldsymbol{m})^t \Gamma_{\boldsymbol{y}}^{-1} (\boldsymbol{y} - A\boldsymbol{m}).$$

In our problem, we may linearize the system model at an estimated parameter vector. Given motion $\boldsymbol{m}$ and structure $\boldsymbol{x}$, the image projection vector is denoted by

$$\boldsymbol{h}(\boldsymbol{m}, \boldsymbol{x}) \triangleq \left(\boldsymbol{h}_1^t, \boldsymbol{h}_1'^t, \boldsymbol{h}_2^t, \boldsymbol{h}_2'^t, \cdots, \boldsymbol{h}_n^t, \boldsymbol{h}_n'^t\right)^t.$$

With the best $\boldsymbol{x}$ directly computed from $\boldsymbol{u}$ and $\boldsymbol{m}$ by $\boldsymbol{y}(\boldsymbol{u}, \boldsymbol{m})$, the residual vector in the image plane is

$$\boldsymbol{f}(\boldsymbol{u}, \boldsymbol{m}) = \boldsymbol{h}(\boldsymbol{m}, \boldsymbol{y}(\boldsymbol{u}, \boldsymbol{m})) - \boldsymbol{u} \tag{4.4}$$

where $\boldsymbol{f}(\boldsymbol{m})$ is a $(4n)$-D vector for n point correspondences (we drop the variable $\boldsymbol{u}$ for simplicity). Given an estimated $\boldsymbol{m}$, $\boldsymbol{m}_i$, expanding $\boldsymbol{f}(\boldsymbol{m})$ at $\boldsymbol{m}_i$ yields

$$\boldsymbol{f}(\boldsymbol{m}) \approx \boldsymbol{f}(\boldsymbol{m}_i) + \frac{\partial \boldsymbol{f}(\boldsymbol{m}_i)}{\partial \boldsymbol{m}}(\boldsymbol{m} - \boldsymbol{m}_i). \tag{4.5}$$

Denoting $J_i \triangleq \frac{\partial \boldsymbol{f}(\boldsymbol{m}_i)}{\partial \boldsymbol{m}}$, (4.5) may be rewritten as

$$-\boldsymbol{f}(\boldsymbol{m}_i) + J_i \boldsymbol{m}_i = J_i \boldsymbol{m} - \boldsymbol{f}(\boldsymbol{m}) + o(||\boldsymbol{m} - \boldsymbol{m}_i||).$$

Denoting the left-hand side as $\boldsymbol{y}_i$ and J_i as A_i, we have

$$\boldsymbol{y}_i = A_i \boldsymbol{m} - \boldsymbol{f}(\boldsymbol{m}) + o(||\boldsymbol{m} - \boldsymbol{m}_i||). \tag{4.6}$$

Neglecting the higher order term $o(||\boldsymbol{m} - \boldsymbol{m}_i||)$, (4.6) is of the form of (4.2). The "noise" term $-\boldsymbol{f}(\boldsymbol{m})$ corresponds to the noise in the coordinates of the image points. Instead of assuming independent Gaussian noise, we just assume that the noises are uncorrelated and have a zero mean and equal variance. By the Gauss-Markov Theorem, the unbiased linear minimum variance estimator is the one that minimizes $||f(\boldsymbol{m})||$ based on the locally linearized equation of (4.6), neglecting the higher order terms. After the iterations correctly converge, the converged point $\boldsymbol{m}_i$ is not far from the true solution if noise is not very large. Therefore, $o(||\boldsymbol{m} - \boldsymbol{m}_i||)$ is small, and (4.6) is a good linear approximation of the system. The nonzero $o(||\boldsymbol{m} - \boldsymbol{m}_i||)$ accounts for the nonlinear nature of the problem. If the convergence occurs far from the true solution, e.g., when the iteration is stuck at a local minimum, $o(||\boldsymbol{m} - \boldsymbol{m}_i||)$ is generally large, and the linearized model does not well characterize the system.

The objective of minimizing the image errors was introduced in Section III-A when maximum likelihood estimation of Gaussian noise distribution is investigated. Here, we discussed the optimality of this objective for general noise distribution. Therefore, it can be expected that minimizing the image error leads to good estimates for other noise distributions, e.g., the uncertainty polyhedron model introduced in Section III-B. Since minimizing the image error is easier to implement than the optimal solution for the uncertainty polyhedron model and is computationally less expensive, it is recommended for general applications where the exact noise distribution is unknown.

V. Error Estimation and Performance Bounds

Further, we need to investigate the following two issues:

1) How can we assess the accuracy of the solutions?
2) What is the theoretical bound of the performance? How close can the performance of the algorithm approach the bound?

A. Error Estimation

The error estimation problem has been discussed for linear algorithms in [55]. There, the aim is basically to estimate errors in the least-squares solution of a linear system $A\boldsymbol{m} = \boldsymbol{y}$, where both matrix A and $\boldsymbol{y}$ are contaminated by noise. The problem here is simpler since only $\boldsymbol{y}$ is contaminated by noise. The reliability of the solution depends not only on noise level but also on the structure of the scene, motion parameters, and the parameters of sensor system. Different components of the motion parameters may have quite different accuracies [55]. Therefore, a method for automatic error estimation is very useful for the problem here.

The minimum variance estimation discussed above leads to a method for estimating errors in the estimates. By the Gauss-Markov Theorem, the covariance matrix of the error vector $\hat{\boldsymbol{m}} - \boldsymbol{m}$ is given by

$$\Gamma_{\hat{\boldsymbol{m}}} \triangleq \mathbf{E}(\hat{\boldsymbol{m}} - \boldsymbol{m})(\hat{\boldsymbol{m}} - \boldsymbol{m})^t = \left(A^t \Gamma_{\boldsymbol{y}}^{-1} A\right)^{-1}.$$

For the nonlinear problem investigated here, the matrix A corresponds to

$$J = \frac{\partial \boldsymbol{f}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{m}})}{\partial \boldsymbol{m}}$$

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

$$\Gamma_{\hat{\boldsymbol{m}}} = \mathbf{E}(\hat{\boldsymbol{m}} - \boldsymbol{m})(\hat{\boldsymbol{m}} - \boldsymbol{m})^t = \sigma^2 \left(J^t J\right)^{-1}. \tag{5.1}$$

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

$$\operatorname{trace}\{\Gamma_{\hat{\boldsymbol{m}}}\} = \mathbf{E}(\hat{\boldsymbol{m}} - \boldsymbol{m})^t(\hat{\boldsymbol{m}} - \boldsymbol{m}) = \mathbf{E}||\hat{\boldsymbol{m}} - \boldsymbol{m}||^2.$$

Since the optimization discussed in Section IV does not require knowledge about exact noise distribution, the error estimation discussed here does not either.

The elements of J are partial derivatives. The partial derivatives can also be estimated by finite differences, which is easier to program. We have implemented both analytical and finite difference versions. The estimated errors showed only negligible differences between those two versions.

The motion parameters can be represented in many ways. Sometimes, one needs to know the errors in terms of the required representation. Generally, for a representation $\boldsymbol{m}' = \boldsymbol{g}(\boldsymbol{m})$, we have

$$\hat{\boldsymbol{m}}' - \boldsymbol{m}' \simeq \frac{\partial \boldsymbol{g}(\hat{\boldsymbol{m}})}{\partial \boldsymbol{m}}(\boldsymbol{m} - \hat{\boldsymbol{m}}).$$

Therefore, with $\Gamma_{\hat{\boldsymbol{m}}}$, the covariance matrix of the error vector of $\hat{\boldsymbol{m}}'$ can be estimated by

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

B. Performance Bounds

Bias and covariance of error provide two measures of the quality of an estimator. Suppose $\hat{\boldsymbol{m}}$ is an estimator of $\boldsymbol{m}$, its expected mean is $\boldsymbol{b}(\boldsymbol{m})$, $\mathbf{E}\hat{\boldsymbol{m}} = \boldsymbol{b}(\boldsymbol{m})$, and its covariance matrix is $C_{\hat{\boldsymbol{m}}}$. We have

$$\Gamma_{\hat{\boldsymbol{m}}} = C_{\hat{\boldsymbol{m}}} + B_{\hat{\boldsymbol{m}}} \tag{5.2}$$

where $\Gamma_{\hat{\boldsymbol{m}}}$ is the correlation matrix of the error vector $\hat{\boldsymbol{m}} - \boldsymbol{m}$: $\Gamma_{\hat{\boldsymbol{m}}} \triangleq \mathbf{E}(\hat{\boldsymbol{m}} - \boldsymbol{m})(\hat{\boldsymbol{m}} - \boldsymbol{m})^t$, and $B_{\hat{\boldsymbol{m}}} = (\boldsymbol{b}(\boldsymbol{m}) - \boldsymbol{m})(\boldsymbol{b}(\boldsymbol{m}) - \boldsymbol{m})^t$. Equation (5.2) implies

$$\operatorname{trace}\{\Gamma_{\hat{\boldsymbol{m}}}\} = \operatorname{trace}\{C_{\hat{\boldsymbol{m}}}\} + \operatorname{trace}\{B_{\hat{\boldsymbol{m}}}\}. \tag{5.3}$$

Therefore, if the estimator is unbiased, the trace of the covariance matrix of the estimator directly gives the expected error. Otherwise, it just indicates a part of the expected error since both terms on the right-hand side of (5.3) are nonnegative. In our case, it is very difficult to get the bias of an estimator in an analytical form. However, the bias is usually small. If we have a lower bound on $C_{\hat{\boldsymbol{m}}}$ for any unbiased estimator, then according to (5.2), this bound is a lower error bound for the unbiased estimator.

Suppose $\boldsymbol{m}$ is a parameter vector of probability density $p(\boldsymbol{u}, \boldsymbol{m})$. $\hat{\boldsymbol{m}}$ is an estimator of $\boldsymbol{m}$ based on measurement $\boldsymbol{u}$ with $\mathbf{E}\hat{\boldsymbol{m}} = \boldsymbol{b}(\boldsymbol{m})$. Letting $\boldsymbol{y}^t = \frac{\partial \ln p(\boldsymbol{u}, \boldsymbol{m})}{\partial \boldsymbol{m}}$, define $F = \mathbf{E}\boldsymbol{y}\boldsymbol{y}^t$, where matrix F is called the Fisher information matrix. Denote

$$B = \frac{\partial \boldsymbol{b}(\boldsymbol{m})}{\partial \boldsymbol{m}}.$$

Then, the error covariance matrix is bounded as

$$\mathbf{E}(\hat{\boldsymbol{m}} - \boldsymbol{b}(\boldsymbol{m}))(\hat{\boldsymbol{m}} - \boldsymbol{b}(\boldsymbol{m}))^t \geq BF^{\dagger}B^t \tag{5.4}$$

where the inequality means that the larger side minus the smaller side is a nonnegative definite matrix. This lower bound is called Cramér-Rao bound. $F^{\dagger}$ is the pseudo inverse of F. If F is invertible, the pseudo inverse is the same as inverse (see [32], [35], [36], and [24] for discussions on pseudo inverse). The equality (5.4) holds if and only if

$$\hat{\boldsymbol{m}} - \boldsymbol{b}(\boldsymbol{m}) = BF^{\dagger} \left(\frac{\partial \ln p(\boldsymbol{u}, \boldsymbol{m})}{\partial \boldsymbol{m}} \right)^t$$

almost everywhere.

The proof of the Cramér-Rao bound can be found in [37] and [58], in [41] for the case $B = I$, and in [8] for the scaler case. Appendix B presents our alternative proof, which seems simpler than the existing proofs.

As we discussed above, although we do not know the actual bias of our estimator, we can compare the actual error with the Cramér-Rao bound for an unbiased estimator.

For an unbiased estimator $\hat{\boldsymbol{m}}$ with identically independently distributed zero mean Gaussian noise added to true image projections, the Cramér-Rao bound gives

$$\Gamma_{\hat{\boldsymbol{m}}} \geq F^{-1} = \sigma^2 (J^t J)^{-1} \tag{5.5}$$

where, using the notation in (4.4)

$$J = \frac{\partial \boldsymbol{f}(\boldsymbol{u}, \boldsymbol{m})}{\partial \boldsymbol{m}}.$$

More generally, if the noise has a covariance matrix C, then the Fisher information matrix is given by $F = J^t C^{-1} J$.

When the minimum attainable variance is larger than the Cramér-Rao bound, other tighter bounds can be derived. For example, the Bhattacharyya bound gives another lower bound of covariance [58], [41], [45]. In fact, the Cramér-Rao bound is actually a special case of the Bhattacharyya bound. Since the Bhattacharyya bound involves higher order derivatives of probability density, the computation is more involved. If the actual errors are close to the Cramér-Rao bound (this is true in the experiments presented in Section VIII-H), the more general Bhattacharyya bound is obviously very close to the Cramér-Rao bound.

In Section VII, the 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.

It is interesting to compare the expressions of error estimation and the Cramér-Rao bound. The estimated error in (5.1) looks similar to the bound in (5.5). However, they are very different. Matrix J in (5.1) is evaluated with the estimated $\boldsymbol{m}$ and the noise-contaminated observation $\boldsymbol{u}$, whereas J in (5.5) is evaluated with the true $\boldsymbol{m}$ and noise-free $\boldsymbol{u}$. In fact, the estimated errors indicate the expected amount of perturbation of solution, away from the *current* solution, which are caused by the amount of perturbation away from the *actual* observations. If the performance of an algorithm is so poor that the estimated parameters have large errors (e.g., a false local minimum), the matrix J does not well characterize the actual system since it is evaluated with bad parameters. In this case, the estimated errors may significantly underestimate the actual errors. Therefore, for a nonlinear problem, a correct convergence is important to the reliability of error estimation. On the other hand, the Cramér-Rao bound is independent of *actual algorithms* and *actual values of noise*.

VI. Computational Considerations

For our problem, the optimizations discussed in the previous sections are nonlinear. Computationally, we need reliable schemes to ensure that the global optimal solution can be reached. This is, in fact, one of the most important issues for any iterative method for nonlinear problems. According to our experience, as discussed in Section VIII-A, in most cases, standard iterative numerical methods that start with a "zero" initial guess do not converge to the correct solution to our problem. In this section, we investigate how to compute the optimal solution reliably and efficiently.

A. A Two-Step Approach

A two-step approach is proposed here. First, a linear algorithm that gives a closed-form solution is applied. Then, in the second step, this solution is used as an initial guess solution for an iterative algorithm, which improves the initial guess to minimize an objective function. This two-step approach has the following advantages:

1) A solution is generally guaranteed. The linear algorithm always gives a solution, provided that degeneracy does not occur [24]. Unless the noise level is very high, this solution is close to the true one. As long as the initial guess is within the convergent region to a globally optimal point, iteration leads to the optimal solution.
2) The approach allows flexible design of the objective functions. If a good initial guess is available, the design of the nonlinear algorithm can emphasize the stability of the derived estimates, and the objective function can be chosen with more flexibility.
3) The approach yields reliable solutions. The linear algorithms use only the epipolar constraint, and therefore, the solution is sensitive to noise, and the reliability of solutions varies with motion types. The optimization in the second step employs more global constraints and achieves significant improvements over the first step.
4) The computation is faster than straight iterative methods that start with a "zero" initial guess. Generally, a linear algorithm is fast, and a nonlinear algorithm is slow. When a linear algorithm is followed by a nonlinear algorithm, the amount of computation is not simply equal to the sum of those needed by each algorithm individually. Since the linear algorithm provides a good initial guess, the time taken by the nonlinear algorithm to reach a solution is greatly reduced.

In order for such a two-step approach to be successful, the initial guess provided by the linear algorithm must be good enough to fall into the convergence area that leads to the global minimum of the selected objective function. As to whether such a requirement is satisfied in our case, we will examine our data of experiments in Section VII.

B. Space Decomposition Using Motion-Structure Dependency

Equation (3.2) involves both motion parameters and 3-D position of every feature point. The maximum is over all the possible motion parameters and scene structures. The parameter space for iteration is huge, and computation is very expensive. However, we do not have to iterate on the structure of the scene.

In fact, given motion parameters $\boldsymbol{m}$, the structure $\boldsymbol{x}$ that minimizes the value of (3.2) can be estimated analytically, that is, we can compute

$$\min_{\boldsymbol{x}} \left\{ ||\boldsymbol{u}_i - \boldsymbol{h}_i(\boldsymbol{m},\boldsymbol{x})||^2 + ||\boldsymbol{u}_i' - \boldsymbol{h}_i'(\boldsymbol{m},\boldsymbol{x})||^2 \right\} \triangleq g_i(\boldsymbol{m}) \tag{6.1}$$

from a given $\boldsymbol{m}$. In fact

$$\min_{\boldsymbol{m},\boldsymbol{x}} \left\{ \sum_{i=1}^{n} \left(||\boldsymbol{u}_i - \boldsymbol{h}_i(\boldsymbol{m},\boldsymbol{x})||^2 + ||\boldsymbol{u}_i' - \boldsymbol{h}_i'(\boldsymbol{m},\boldsymbol{x})||^2 \right) \right\}$$

$$= \min_{\boldsymbol{m}} \left\{ \sum_{i=1}^{n} \min_{\boldsymbol{x}} \left\{ ||\boldsymbol{u}_i - \boldsymbol{h}_i(\boldsymbol{m},\boldsymbol{x})||^2 + ||\boldsymbol{u}_i' - \boldsymbol{h}_i'(\boldsymbol{m},\boldsymbol{x})||^2 \right\} \right\} = \min_{\boldsymbol{m}} \sum_{i=1}^{n} g_i(\boldsymbol{m}).$$

Therefore, computationally, structure $\boldsymbol{x}$ will not be included in the parameter space of iteration. Given an $\boldsymbol{m}$, $\boldsymbol{x}$ can be computed directly. This drastically reduces the amount of computation. Otherwise, it is computationally extremely expensive to iterate on this huge $(\boldsymbol{m},\boldsymbol{x})$ space (iterations on n points need $(3n+5)$-D parameter space). Since the optimal structure $\boldsymbol{x}$ can be determined from motion parameters, we can exclude $\boldsymbol{x}$ from the notation for parameters to be estimated, that is, symbolically, the parameters to be determined are just $\boldsymbol{m}$.

To derive the closed-form expression for $\boldsymbol{x}$ that gives $g_i(\boldsymbol{m})$ in (6.1), we use the following methods. From motion parameter vector $\boldsymbol{m}$ and the observed projections of point i, the two observed projection lines are determined. These two observed projection lines do not intersect in general (see Fig. 2). If the true 3-D point is on the observed projection line of the first image, the discrepancy $||\boldsymbol{u}_i - \boldsymbol{h}_i(\boldsymbol{m},\boldsymbol{x})||^2$ is equal to zero, but $||\boldsymbol{u}_i' - \boldsymbol{h}_i'(\boldsymbol{m},\boldsymbol{x})||^2$ is generally not. If the true 3-D point is on the other observed projection line, $||\boldsymbol{u}_i' - \boldsymbol{h}_i'(\boldsymbol{m},\boldsymbol{x})||^2$ is equal to zero, whereas $||\boldsymbol{u}_i - \boldsymbol{h}_i(\boldsymbol{m},\boldsymbol{x})||^2$ is not. Given the motion parameters, we need to find a 3-D point for each feature point such that the corresponding term $||\boldsymbol{u}_i - \boldsymbol{h}_i(\boldsymbol{m},\boldsymbol{x})||^2 + ||\boldsymbol{u}_i' - \boldsymbol{h}_i'(\boldsymbol{m},\boldsymbol{x})||^2$ is minimized. Obviously, under a normal configuration, the point lies in the shortest line segment L that connects the two observed projection lines (see Fig. 2) because otherwise, the perpendicular projection of a 3-D point onto L is better than the 3-D point. An exact solution of the optimal point requires solving a fourth-order polynomial equation. It can be shown that using a reasonable approximation, we can get a closed-form solution. The optimal point is generally not far from the midpoint of the line segment L unless the distance to the object and the viewing angle differ a lot for two images. In simulations, only a small performance difference is observed if the midpoint is used as the optimal point instead of the analytical solution. For computational efficiency, we may just use the midpoint of the line L as an approximated optimal point.

For both maximum likehood estimator under Gaussian noise and minimum variance estimator for general unknown noise, we minimize image error, which is equivalent to the following minimization:

$$\min_{\boldsymbol{m}} ||\boldsymbol{f}(\boldsymbol{u},\boldsymbol{m})||. \tag{6.2}$$

Denoting $J_i \triangleq \frac{\partial \boldsymbol{f}(\boldsymbol{m}_i)}{\partial \boldsymbol{m}}$, where $\boldsymbol{m}_i$ is the ith estimate of $\boldsymbol{m}$, and using the Levenberg-Marquardt (L-M) method [22], [27], [34], [26], we get a sequence of approximation to a minimum solution

$$\boldsymbol{m}_{i+1} = \boldsymbol{m}_i + \left(D_i + J_i^t J_i\right)^{-1} J_i^t \boldsymbol{f}(\boldsymbol{m}_i)$$

where D_i is a diagonal matrix with nonnegative diagonal elements. The finite difference analogs of the L-M and Gauss

algorithms by Brown [6] does not require an analytical expression of Jacobian J_i.

C. Batch and Sequential Solutions

If all the data acquired are processed simultaneously, the 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 the Kalman filtering, the 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. This section shows that although both types of techniques are mathematically equivalent for linear problems, the performance of the batch techniques is generally better than that of the sequential techniques for nonlinear problems.

1) Linear Systems: In the interest of generality, we first extend the problem to allow the parameters to change, or evolve, when observations are acquired sequentially. The system dynamic model (which is also known as the plant model) is given by

$$\boldsymbol{m}_{k+1} = \Phi_k \boldsymbol{m}_k + \boldsymbol{\eta}_k \tag{6.3}$$

and measurement $\boldsymbol{y}_k$ is determined by the measurement model

$$\boldsymbol{y}_k = A_k \boldsymbol{m}_k + \boldsymbol{\epsilon}_k \tag{6.4}$$

$k = 0, 1, 2, \ldots$, where $\boldsymbol{m}_k$ is the parameter vector (or state) at time k, Φ_k and A_k are matrices, $\boldsymbol{\eta}_k$ and $\boldsymbol{\epsilon}_k$ are random errors with zero means, $\mathbf{E}\boldsymbol{\eta}_i\boldsymbol{\eta}_j^t = \delta_{ij}Q_i$, $\mathbf{E}\boldsymbol{\epsilon}_i\boldsymbol{\epsilon}_j = \delta_{ij}C_i$, and $\boldsymbol{\eta}_i$ and $\boldsymbol{\epsilon}_j$ are all uncorrelated for $i \geq 0$ and $j \geq 0$. ($\delta_{ij} = 0$ for $i \neq j$ and $\delta_{ii} = 1$ for all i.)

The linear time-varying system can be solved naturally and efficiently by a sequential approach, due to the dynamic nature (or time-varying parameters) of system (6.3). A sequential least-squares approach called Kalman filtering [20], [13], [28], [3], [41], [29], [14] is widely used. The formulation of Kalman filtering is relegated into Appendix C, which is initialized by $\hat{\boldsymbol{m}}_{0,-1} = \boldsymbol{a}$ and $P_{0,-1} = \mathbf{E}(\hat{\boldsymbol{m}}_{0,-1} - \boldsymbol{a})(\hat{\boldsymbol{m}}_{0,-1} - \boldsymbol{a})^t$.

The Kalman filtering algorithm can be derived by either probabilistic or deterministic methods. Both types methods are unified under Hilbert space optimization [24], [14]. The Kalman filtering algorithm can also be derived by solving for $\hat{\boldsymbol{m}}_i$ and $\hat{\boldsymbol{\eta}}_i$ that minimizes

$$(\hat{\boldsymbol{m}}_0 - \boldsymbol{a})^t P_{0,-1}^{-1}(\hat{\boldsymbol{m}}_0 - \boldsymbol{a}) + \sum_{i=0}^{n} (\boldsymbol{y}_i - A_i\hat{\boldsymbol{m}}_i)^t C_i^{-1}(\boldsymbol{y}_i - A_i\hat{\boldsymbol{m}}_i) + \sum_{i=0}^{n-1} \hat{\boldsymbol{\eta}}_i^t Q_i^{-1}\hat{\boldsymbol{\eta}}_i \tag{6.5}$$

subject to the constraint $\boldsymbol{m}_{k+1} = \Phi_k\boldsymbol{m}_k + \boldsymbol{\eta}_k$.

Without *a priori* knowledge of the parameters, the batch solution and the sequential solution are the same for linear systems.

2) Nonlinear Systems: However, like most real-world problems, the problem of estimating motion and structure parameters of the scene is nonlinear. A general nonlinear system with time-invariant parameters can be expressed as

$$\boldsymbol{f}(\boldsymbol{h}, \boldsymbol{m}) = \mathbf{0} \tag{6.7}$$

where $\boldsymbol{h}$ is a noise-free observation vector, and $\boldsymbol{m}$ is the parameter to be estimated. With actual noisy observation $\boldsymbol{h}^*$ and estimate $\boldsymbol{m}^*$, expending $\boldsymbol{f}(\boldsymbol{h}, \boldsymbol{m})$ at $(\boldsymbol{h}^*, \boldsymbol{m}^*)$ yields

$$\boldsymbol{f}(\boldsymbol{h}^*, \boldsymbol{m}^*) + \frac{\partial \boldsymbol{f}(\boldsymbol{h}^*, \boldsymbol{m}^*)}{\partial \boldsymbol{h}}(\boldsymbol{h} - \boldsymbol{h}^*) + \frac{\partial \boldsymbol{f}(\boldsymbol{h}^*, \boldsymbol{m}^*)}{\partial \boldsymbol{m}}(\boldsymbol{m} - \boldsymbol{m}^*) \approx \mathbf{0}. \tag{6.8}$$

Letting

$$J = \frac{\partial \boldsymbol{f}(\boldsymbol{h}^*, \boldsymbol{m}^*)}{\partial \boldsymbol{m}}, G = \frac{\partial \boldsymbol{f}(\boldsymbol{h}^*, \boldsymbol{m}^*)}{\partial \boldsymbol{h}} \tag{6.9}$$

$\boldsymbol{y} = -\boldsymbol{f}(\boldsymbol{h}^*, \boldsymbol{m}^*)$, $\mathbf{E} = -G(\boldsymbol{h} - \boldsymbol{h}^*)$, and neglecting higher order terms, (6.8) becomes

$$\boldsymbol{y} = J(\boldsymbol{m} - \boldsymbol{m}*) + \mathbf{E}. \tag{6.10}$$

The vector $\boldsymbol{h} - \boldsymbol{h}^*$ gives the difference between the noise-free observation $\boldsymbol{h}$ and actual observation $\boldsymbol{h}^*$. Therefore, $\boldsymbol{h} - \boldsymbol{h}^*$ corresponds to observation noise. Letting $\mathbf{E}(\boldsymbol{h} - \boldsymbol{h}^*)(\boldsymbol{h} - \boldsymbol{h}^*)^t = Q_h$, we have $\mathbf{E}\mathbf{E}\mathbf{E}^t = GQ_hG^t$.

By the Gauss-Markov Theorem, obtaining the linear minimum variance estimate of $(\boldsymbol{m} - \boldsymbol{m}^*)$ in (6.10) yields

$$\hat{\boldsymbol{m}} = \boldsymbol{m}^* + \left(J^t\left(GQ_hG^t\right)^{-1}J\right)^{-1}\left(GQ_hG^t\right)^{-1}J^t\boldsymbol{y}. \tag{6.11}$$

Since the problem is nonlinear, J is evaluated at the estimated parameter $\boldsymbol{m}^*$. A sequence of approximations of $\boldsymbol{m}$ is obtained by iterations based on (6.11), each iteration replacing $\boldsymbol{m}^*$ by $\hat{\boldsymbol{m}}$. After the iterations converge, the estimate $\hat{\boldsymbol{m}}$ is an approximated minimum variance estimate of $\boldsymbol{m}$ or, equivalently, a weighted nonlinear least-squares approximate solution that minimizes $\boldsymbol{f}(\hat{\boldsymbol{m}})^t\left(GQ_hG^t\right)^{-1}\boldsymbol{f}(\hat{\boldsymbol{m}})$. This is a batch technique for nonlinear systems.

For time-varying parameters, the system model (6.3) should be replaced by

$$\boldsymbol{m}_{k+1} = \Phi_k(\boldsymbol{m}_k) + \boldsymbol{\eta}_k \tag{6.12}$$

where $\Phi_k(\boldsymbol{m}_k)$ is a nonlinear function. In principle, a batch technique can also be applied to nonlinear system with time-varying parameters.

With a typical sequential processing technique (nonlinear Kalman filtering), conventionally, a nonlinear measurement model is assumed:

$$\boldsymbol{y}_k = \boldsymbol{h}_k(\boldsymbol{m}_k) + \boldsymbol{\epsilon}_k \tag{6.13}$$

where $\boldsymbol{h}_k(\boldsymbol{m})$ is a nonlinear function. Equation (6.13) is a special case of (6.7) in that the observation can be explicitly expressed as a function of parameters and measurement errors.

Unlike the case of a linear system, for a nonlinear system, the batch solution and sequential solution are not the same. J in (6.10) for a nonlinear problem corresponds to A in (4.2) for

a linear problem. Although A is constant, J is not. J defined in (6.9) is a function of observation $\boldsymbol{h}^*$ and current estimate $\boldsymbol{m}^*$.

Due to the sequential nature of the Kalman filter, the estimate of $\boldsymbol{m}_k$ is based on the estimate of $\boldsymbol{m}_{k-1}$, its covariance matrix, and a new observation. Once the previous parameter $\boldsymbol{m}_{k-1}$ is estimated, it is not improved after new observations $\{\boldsymbol{y}_i\}$ and $i > k$ are obtained. Let us consider the nonlinear system of (6.13). Using the Kalman filter, the objective function (6.5) becomes

$$(\hat{\boldsymbol{m}}_0 - \boldsymbol{a})^t P_{0,-1}^{-1}(\hat{\boldsymbol{m}}_0 - \boldsymbol{a}) + \sum_{i=0}^{n} (\boldsymbol{y}_i - J_i\hat{\boldsymbol{m}}_i)^t C_i^{-1}(\boldsymbol{y}_i - J_i\hat{\boldsymbol{m}}_i) + \sum_{i=0}^{n-1} \hat{\boldsymbol{\eta}}_i^t Q_i^{-1}\hat{\boldsymbol{\eta}}_i \tag{6.14}$$

subject to $\boldsymbol{m}_{k+1} = \Phi_k(\hat{\boldsymbol{m}}_k) + \boldsymbol{\eta}_k$ and

$$J_i = \frac{\partial \boldsymbol{h}_i\left(\hat{\boldsymbol{m}}_i^{(i)}\right)}{\partial \boldsymbol{m}}$$

where $\hat{\boldsymbol{m}}_i^{(i)}$ is the estimated $\boldsymbol{m}_i$ based on the observations $\{\boldsymbol{y}_k | 0 \le k \le i\}$. To see more clearly what (6.14) implies, assume that the parameter is time invariant $\Phi_k(\hat{\boldsymbol{m}}_k) = \hat{\boldsymbol{m}}_k$, $\boldsymbol{\eta}_k = \boldsymbol{0}$, $C_i = I$, and $P_{0,-1}^{-1} = 0$. Equation (6.14) becomes

$$\sum_{i=0}^{n} ||\boldsymbol{y}_i - J_i\hat{\boldsymbol{m}}||^2 = \sum_{i=0}^{n} ||\boldsymbol{y}_i - \frac{\partial \boldsymbol{h}_i\left(\hat{\boldsymbol{m}}^{(i)}\right)}{\partial \boldsymbol{m}}\hat{\boldsymbol{m}}||^2. \tag{6.15}$$

We can compare (6.15) with (6.10). To improve the performance for nonlinear systems, the Kalman filtering algorithm needs iterations (iterated extended Kalman filter (IEKF) algorithm) at each observation $\boldsymbol{y}_i$: After $\boldsymbol{m}$ is estimated from $\boldsymbol{y}_i$, it is used to evaluate J_i. $\boldsymbol{y}_i$ and the improved J_i in turn give a new estimate of $\boldsymbol{m}$. Such an iteration is carried on until no improvement for $\boldsymbol{m}$ occurs. Then, $\hat{\boldsymbol{m}}^{(i)}$ is determined. For small i, $\hat{\boldsymbol{m}}^{(i)}$ has a large error since just $i+1$ observations are available. Therefore, J_i evaluated at $\hat{\boldsymbol{m}}^{(i)}$ 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 updated later when new observations are collected. They are included in the objective function (6.15), further preventing the estimated parameter $\boldsymbol{m}$ from approaching the correct parameters while new data are obtained. The more nonlinear $\boldsymbol{h}(\boldsymbol{u}, \boldsymbol{m})$ is in terms of $\boldsymbol{m}$, the worse J_i is. Therefore, sequential methods generally are not suitable to be used to solve a highly nonlinear equation from an arbitrary initial guess for $\boldsymbol{m}$.

The performance of the IEKF algorithm also depends very much on the initial guess solution $\boldsymbol{a}$ and the initial covariance matrix $P_{0,-1}$. Table I lists the effects of initial condition $\hat{\boldsymbol{m}}_{0,-1}$ and the associated covariance matrix $P_{0,-1}$ on the estimated parameter $\hat{\boldsymbol{m}}_k$ for a nonlinear time-varying system. In all cases, the covariance matrix P_k is always small, that is, P_k may significantly underestimate the errors of the estimated parameters.

TABLE I
CHARACTERISTICS OF IEKF FOR NONLINEAR SYSTEMS

$\hat{\boldsymbol{m}}_{0,-1}$	$P_{0,-1}$	$\hat{\boldsymbol{m}}_k$	P_k
bad	large	divergent	small
bad	small	not improved	small
good	large	divergent	small
good	small	improved	small

The batch method treats J_i differently. For the model discussed above, the objective of the batch solution is to minimize

$$\sum_{i=0}^{n} ||\boldsymbol{y}_i - J_i\hat{\boldsymbol{m}}||^2 = \sum_{i=0}^{n} ||\boldsymbol{y}_i - \frac{\partial \boldsymbol{h}_i(\hat{\boldsymbol{m}})}{\partial \boldsymbol{m}}\hat{\boldsymbol{m}}||^2 \tag{6.16}$$

through iterations. We can compare (6.16) with (6.15). In each iteration, the estimated parameters are modified based on all observations (unlike the sequential algorithm, which modifies the parameters with each observation). Therefore, such a modification is more reliable. Further, J_i is always updated through iterations. As a local minimum is reached, all J_i's are evaluated at this final solution.

With the Kalman filter, divergence is said to occur when the error covariance matrix computed by the filter becomes unjustifiably small compared with the actual error in the estimate. Divergence of the Kalman filter has been observed for nonlinear systems, and it has been attributed to system nonlinearity [40], [12], [3], [41], [29], [5]. Although it is known that states and the associated covariance are no longer sufficient statistics in a nonlinear system with Gaussian noise, the underlying problems of Kalman filtering for nonlinear systems, with Gaussian or nonGaussian noise, have not been fully analyzed. Our discussion here focuses on the reasons for the poorer performance, and (6.15) and (6.16) display a fundamental difference between the sequential and batch solutions. It is worth mentioning that the conventional Gaussian noise assumption is not required for our discussion.

In summary, the divergence and poorer performance of Kalman filtering, compared with batch approaches, are consequences of the following: a) Fluctuations of parameters when updating using early observations cause system matrix J_i to be evaluated at bad parameters, even if the initial parameters are good; b) system matrix J_i is not updated using newer observations $\boldsymbol{y}_k$, $k > i$. The performance difference between a sequential and a batch approach is especially large for highly nonlinear systems.

VII. EXPERIMENTAL RESULTS

To further verify the analyses presented above and demonstrate the performances of the algorithms, experiments using simulated data and real-world images have been conducted.

For the simulated data, the focal length is one unit. The image frame forms a $s \times s$ square. The field of view is then determined by the image size s and the unit focal length. Unless stated otherwise, $s = 0.70$, and 12-point correspondences are used. The object points are generated randomly and uniformly between depths 6 and 16. The image coordinates of the points are digitized according to the spatial

resolution of the camera. If the resolution is m by m, the horizontal and vertical coordinates each have uniformly spaced m levels. The image coordinates are rounded off to the nearest levels before they are used by the motion estimation algorithm. Although the noise here is simulated by digitization noise, it may represent other kinds of noise. For example, additive noise can be simulated by a reduced resolution. (Our experiments on real images have indicated that the error variance of the points given by the matching algorithm are generally not larger than the quantization noise of a 256 × 256 image.)

All errors shown in this section are relative, except for those with real images. Relative error of a matrix, or vector, is defined by the Euclidean norm of the error matrix, or vector, divided by the Euclidean norm of the correct matrix, or vector, respectively. The linear algorithm presented in [55] was used to generate initial guesses.

The subroutine ZXSSQ in the IMSL library is designed to find the minimum of the sum of squares of multivariate nonlinear functions using a finite difference Levenberg-Marquardt algorithm. This subroutine was used for the batch nonlinear minimization in our experiments presented here.

A. Whether a Good Initial Guess Is Necessary

First, to investigate whether the initial guess provided by the linear algorithm indeed helps the second optimization step, a fixed guess is used as the initial guess. The fixed initial guess has a zero rotation angle and a translation vector of $(1, 1, 1)$. The image resolution is 256 × 256, and the sign reversal for the translation and rotation angle is performed when the iteration does not converge. Different motion parameters are selected randomly. Among 36 examples with 12 point correspondences, 16 of them do not converge, or they converge to a wrong answer. A wrong answer means that the error of translation is larger than 100%, or the error of rotation matrix is larger than 50%. Remaining cases give correct solutions with the translation error less than 10% and the rotation error less than 5%. This shows that a good initial guess is necessary to ensure a correct solution.

If the noise is moderate (resolution is 128 × 128 or higher), the initial guess provided by the linear algorithm [49] in the first step is generally good enough to ensure correct convergence in the second step. With a resolution of 64 × 64 or lower, very good estimation of motion parameters is not possible since the the Cramér-Rao bound is already large. However, if one insists on getting something from this very low resolution, the initial guess given by the linear algorithm is severely contaminated by noise, and the iteration might not correctly converge. Thus, a search can be conducted on a coarse grid in the parameter space, and the initial guess is selected as the one at which the objective function reaches the minimum among all the grid points.

B. Minimizing the Epipolar Error Versus Minimizing the Image Error

After the solution is obtained by the linear algorithm, the solution can be improved by using the constraint on E to minimize the sum of a weighted version of (2.5):

$$\sum_{i=1}^{n} \frac{\left((\boldsymbol{X}_i')^t(\boldsymbol{T}_s \times R\boldsymbol{X}_i)\right)^2}{\sigma^2\left(||R^t(\boldsymbol{T}_s \times \boldsymbol{X}_i')||_{z=0}^2 + ||\boldsymbol{T}_s \times R\boldsymbol{X}_i||_{z=0}^2\right)}$$

where the weight has been presented in (2.6). This is called the epipolar improvement, and the above expression is called the epipolar error. By doing this, the constraint in matrix E is taken into account since E appears as a decomposed form in the above expression. However, the effect of the scene structure discussed in Sections II-D and E is not used.

Fig. 9 compares the errors from the linear algorithm, the epipolar improvement, and minimizing the image error, respectively, using different fields of view. First, we see that minimizing the image error brings about significant improvement over the linear algorithm: Error reduction is about a factor 2 for rotation and a factor of 4 to 8 for translation. Next, we consider the difference between the epipolar improvement and minimizing the image error in terms of the accuracy of the solutions. For longitudinal translation (orthogonal to the image plane), both methods give very similar errors since the solutions of the linear algorithm are already very good [55], and the epipolar constraint is capable of disambiguating motion in this case. The performance shows a difference for unstable lateral translations (which are parallel to the image plane). As indicated in Fig. 9, the epipolar improvement over the linear algorithm is obviously significant, but the errors after minimizing the image errors are further significantly smaller than those after minimizing the epipolar errors, especially for small fields of view.

C. Sequential Versus Batch Solutions

Minimizing the image error in (3.3) is a nonlinear problem. We compute batch solutions using the L-M method. The sequential solution is obtained by the IEKF. Iterations are performed for each point correspondence (four observations, i.e., rank-four update) to improve the performance of the regular IEKF (which just iterates on one observation).

Fig. 10 shows typical sequences of IEKF sequential updating. In Fig. 10(a), with a very good initial guess and a relatively small diagonal covariance matrix $P_{0,-1}$, the IEKF improves the initial guess. In Fig. 10(b), a large diagonal covariance matrix $P_{0,-1}$ is used. The final errors are significantly larger than those in Fig. 10(a) because the Kalman filter updates the estimates without much *a priori* information due to a large initial covariance. A few early noise-contaminated observations cause premature updating of the estimated parameters, which causes J_k to be evaluated at deteriorated parameters. In Fig. 10(c), the IEKF diverges from a zero initial guess and a large diagonal initial covariance $P_{0,-1}$ (if a small $P_{0,-1}$ is used, the errors will remain almost the same as that of the initial guess).

Next, the performances of the linear algorithm, a sequential algorithm (IEKF) and a batch algorithm (the L-M method) are compared in Fig. 11. Both the IEKF and the L-M method use the solutions of the linear algorithm as initial guesses.

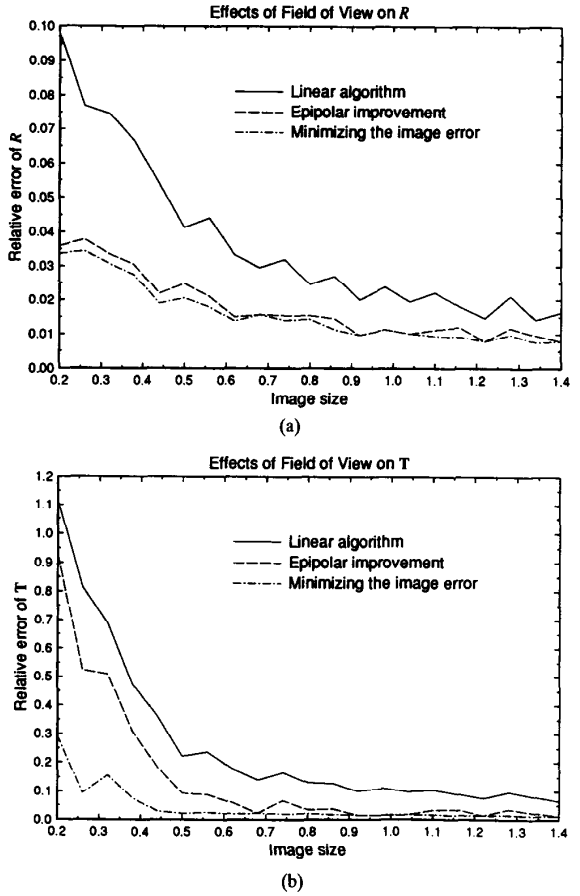


Fig. 9. Epipolar improvement versus minimizing the image errors. Rotation axis: $(1,1,1)$; rotation angle: 1o; translation: $(t_x,0,0)$, where t_x varies linearly from 0.1 (for image size 0.2) to 0.7 (for image size 1.4). One-hundred random trials: (a) Relative errors of rotation matrix R; (b) relative errors of translation vector T.

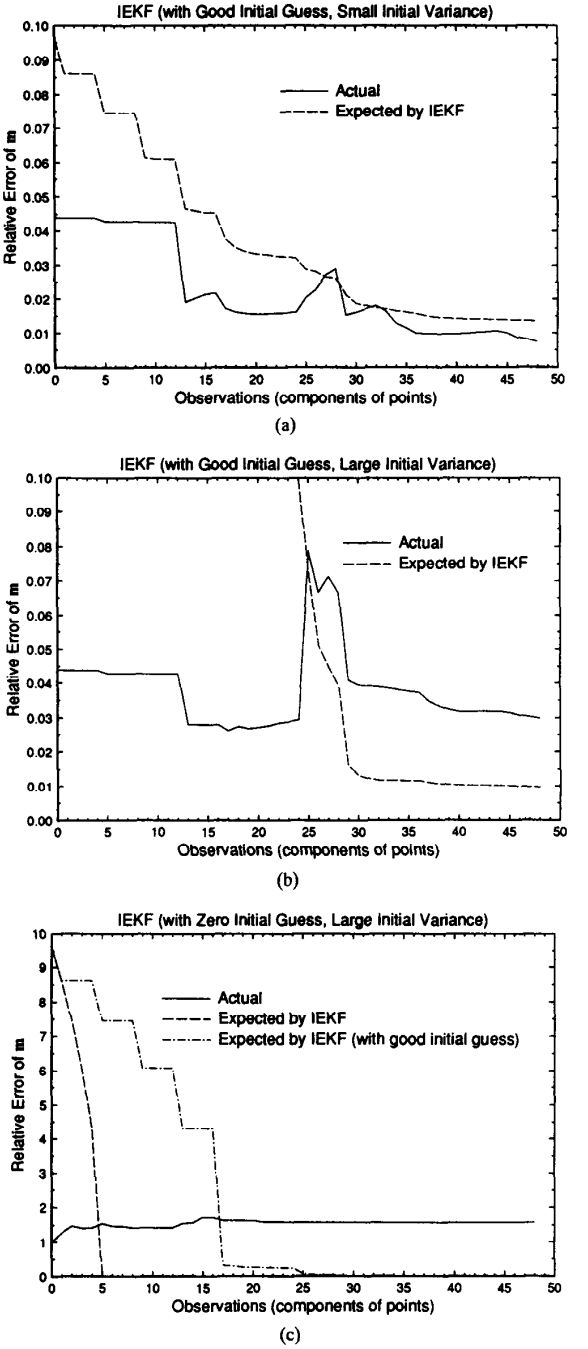
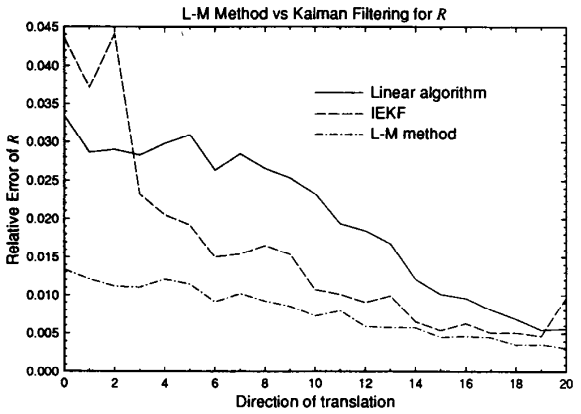


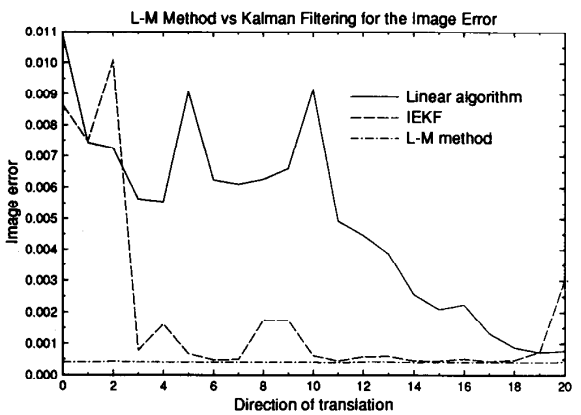
Fig. 10. IEKF for the nonlinear problem: a sample sequence. Rotation axis: $(1,1,1)$; rotation angle: 3o; translation vector: $(1.732,1.732,-1.732)$. "Expected by IEKF" denotes the error predicted by the covariance matrix P_k: (a) With good initial guesses, generated by the linear algorithm, and a small diagonal $P_{0,-1}$; (b) with the same good initial guesses as in (a) and a large $P_{0,-1}$. For a clearer display of the other values, the covariance of IEKF is not shown here for the cases with less than 25 observations and is shown in (c) instead under "Part of expected by IEKF (with good initial guess)"; (c) with zero initial guesses and a large $P_{0,-1}$.

The sequential algorithm improves over linear algorithm for most of cases. However, for some cases, the results are worse than the initial guesses (see some average errors in Fig. 11). The initial covariance matrix $P_{0,-1}$ of the IEKF is carefully selected (same as that in Fig. 10(a)). A larger $P_{0,-1}$ will result in more divergent cases, and a smaller one will impede parameter improvement. In contrast, the image error in Fig. 11(d) shows that the batch method finds the minimum very consistently, but the IEKF does not always find the minimum. Sometimes, the IEKF diverges. From Fig. 11, we can see that the batch optimization significantly outperforms the IEKF sequential algorithm.

With different resolutions, the improvement of the batch optimization over the linear algorithm is shown in Fig. 12. As can be seen, the motion error reduction is a factor of 2 to 3. It can also be seen that the average image errors after the batch minimization are always about equal to the standard deviation of the actual noise in the image coordinates. This is true even for extremely low resolution (32×32). This implies that the global minimum solutions can be consistently reached by the batch method, and the solutions of the linear algorithm are good enough to be used as initial guesses.

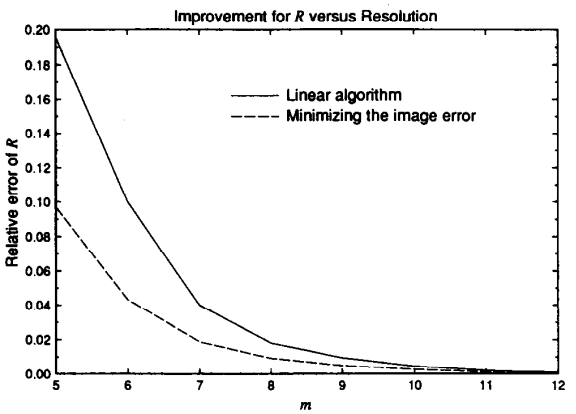


(a)

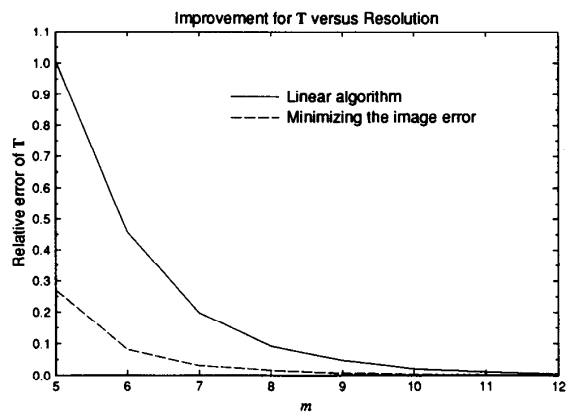


(b)

Fig. 11. Relative errors of the linear algorithm, batch solution (the L-M method), and sequential solution (IEKF). Rotation axis: $(1, 1, 1)$; rotation angle: 3o. For a horizontal index from 0 to 20, the direction of translation changes from $(1, 0, 0)$ to $(0, 0, 1)$ in the xz plane at evenly spaced 21 steps. The length of the translation vector is equal to 2.1 units. 100 random trials: (a) R; (b) image errors.



(a)



(b)

Fig. 12. Improvement of the batch optimization over the linear algorithm versus image resolutions. The simulated image has $2m \times 2m$ pixels. Image size: 1; rotation axis: $(1, 1, 1)$; rotation angle: 5o; translation: $(3, 0, 0)$; 100 random trials: (a) R; (b) T.

D. The Uncertainty Polyhedron Model

Fig. 13 presents the comparison between model 1 (the Gaussian distribution) and model 2 (the uncertainty polyhedron distribution). The actual noise is spatial digitization noise. Although model 2 seems more appropriate here than model 1, the errors in motion parameters are very similar between these two models. This implies that the performance of minimizing the image error is not very sensitive to the changes in the assumed noise distribution.

Here, it would be clearer to observe the image errors in Fig. 13(b). The value of the resulting image error from the method of model 2 is virtually equal to the standard deviation of the actual image plane noise, indicating again that the global minimum is consistently reached from the initial guess provided by the linear algorithm.

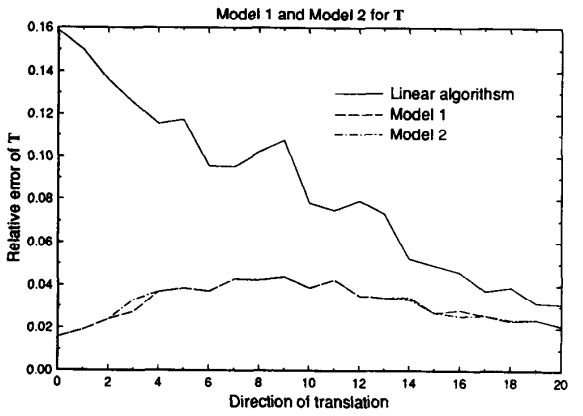
E. Error Estimation and the Cramér-Rao Bound

Fig. 14 shows the average relative errors, the average deviation of the error estimation (which is the absolute difference between the estimated relative errors and the actual relative errors), and the bias of error estimation (which is the average difference between the estimated relative errors and the actual relative errors) over 40 random trials. As can be seen from the figure, the bias is small, and the average deviation is generally around a half of the magnitude of the actual relative errors.

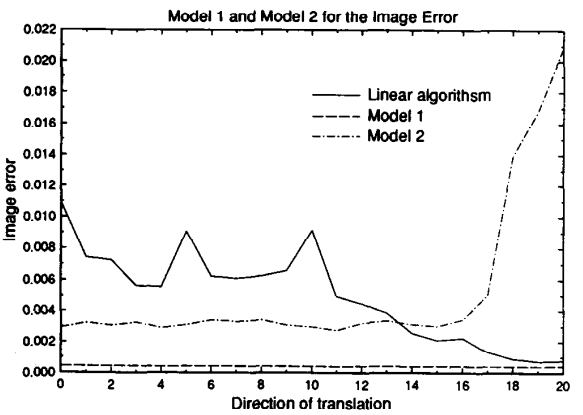
It is worth mentioning that the average deviation of error estimates reflects the deviation of actual errors (which are directly related to standard deviation or variance of the errors). To see the deviation of errors in the solutions of the linear algorithm, refer to the result of error estimation for the linear algorithm (see Fig. 8 of [55]).

Fig. 15 shows the comparison between the actual relative errors and the corresponding Cramér-Rao bound with Gaussian noise. Two types of noise are simulated: Gaussian and uniform (with same variance). As shown in Fig. 15(a) and (b), the actual relative errors in the estimated solutions are very close to the Cramér-Rao bound with Gaussian noise. In other words, the errors of the algorithm are very close to those of a best possible unbiased estimator with Gaussian noise.

By comparing Fig. 15(b) and (c) with Fig. 7(a) and (b) it can be seen that the errors under digitization noise are similar to those under the Gaussian noise with the same variance. This



(a)



(b)

Fig. 13. Relative errors of the linear algorithm: model 1 (minimizing the image error) and model 2 (uncertainty polyhedron). Rotation axis: $(1, 1, 1)$; rotation angle: 3o; for horizontal index from 0 to 20, the direction of translation changes from $(1, 0, 0)$ to $(0, 0, 1)$ in the xz plane at 21 evenly spaced steps. The length of the translation vector is equal to 2.1 units. 100 random trials: (a) T; (b) image error.

implies that the shape of noise distribution function does not significantly affect the actual errors as long as the variance is kept the same. Therefore, it is reasonable to expect, provided that the data do not contain extremely bad data or outliers [19], that minimizing the image error will lead to good performance under general noise, even if the shape of noise distribution function differs significantly from Gaussian.

Fig. 15 also shows the relative absolute bias of the estimates (the norm of the bias matrix, or vector, divided by the norm of the true matrix, or vector). The bias is small relative to the actual errors.

The performance of the algorithm has virtually reached the theoretical lower bounds with Gaussian noise. This fact provides quantitative insight into the impact of errors that may seem negligible, such as digitization errors. From Fig. 15, we know that to give an estimated translation direction with under 2.0% expected error, the variance of the errors in the image points cannot be larger than those of digitization errors of 256×256 images.

We can also see from Figs. 12 and 15 that the error in the estimate is roughly inversely proportional to the image resolution and the number of point correspondences. To reduce the error by a factor of two, for example, one can either double the image resolution (reduce the amount of noise in image point position) or double the number of point correspondences.

F. Inherent Limitation of Small Motion

The restriction of small motion for optical flow-based approaches arises primarily from two facts: 1) Optical flow, by ideal definition, is the projection of 3-D velocity onto the image plane. The formulation of computing optical flow is in terms of velocity (e.g., [17], [15], [33], [21], [16], [47]). 2) The mathematical formulation of computing motion parameters from optical flow is also in terms of motion velocity [7], [60], [1], [48]. However, what is actually observed is the displacement between images. Only in the case of small motion can displacement be approximated by velocity.

Although the restriction of small interframe motion simplifies both computing image matching (optical flow as a result) and computing motion parameters from optical flow (motion velocity as a result), the reliability of the computed motion parameters in the presence of noise is inherently limited. A small amount of motion is easily overridden by the pixel-level errors in the estimated optical flow. Even if the optical flow can be estimated with a subpixel precision, such a subpixel precision does not mean subpixel accuracy since spatial digitization noise coupled with variation and discontinuity of flow field makes smoothing or interpolation less effective in terms of accuracy. The essence here is that with a small motion, the ratio of signal to spatial digitization noise is low since the power of the latter is a constant.

As shown in Fig. 15, our algorithm has essentially reached the theoretical error bound. What about small motions, e.g., those typically used for velocity-based formulation or optical flow? We consider a setup: The image is a unit square with 512×512 pixels. We assume that the image positions of the points are contaminated by additive white Gaussian noise with a variance equal to that of the uniform distribution in the range of ± 1 pixel. The magnitudes of translation are such that the maximum disparities caused by translation in the image plane are 2, 4, 8, and 16 pixels, respectively. The Cramér-Rao bounds on the relative error in the estimated translation, which average over 10 random point sets, are shown in Fig. 16. It can be seen that under a small motion with a 2-pixel maximum disparity (average disparity is roughly 1 pixel), the errors in translation are bounded below by 60%, even when using a large number of points (70). For a small motion with a 4-pixel maximum disparity, the error bound is still large (about 38% with 70 points). Whether or not one uses discrete features or optical flow, the data shown here quantitatively predict the inherent stability problem for motion and structure from small motions.

G. Real-World Images

Our algorithm has been tested on real images. A CCD video camera with roughly 480×500 pixels was used to take two images of each scene at unknown positions. The field of view is roughly equivalent to that of a f=62.5 mm 35-mm camera

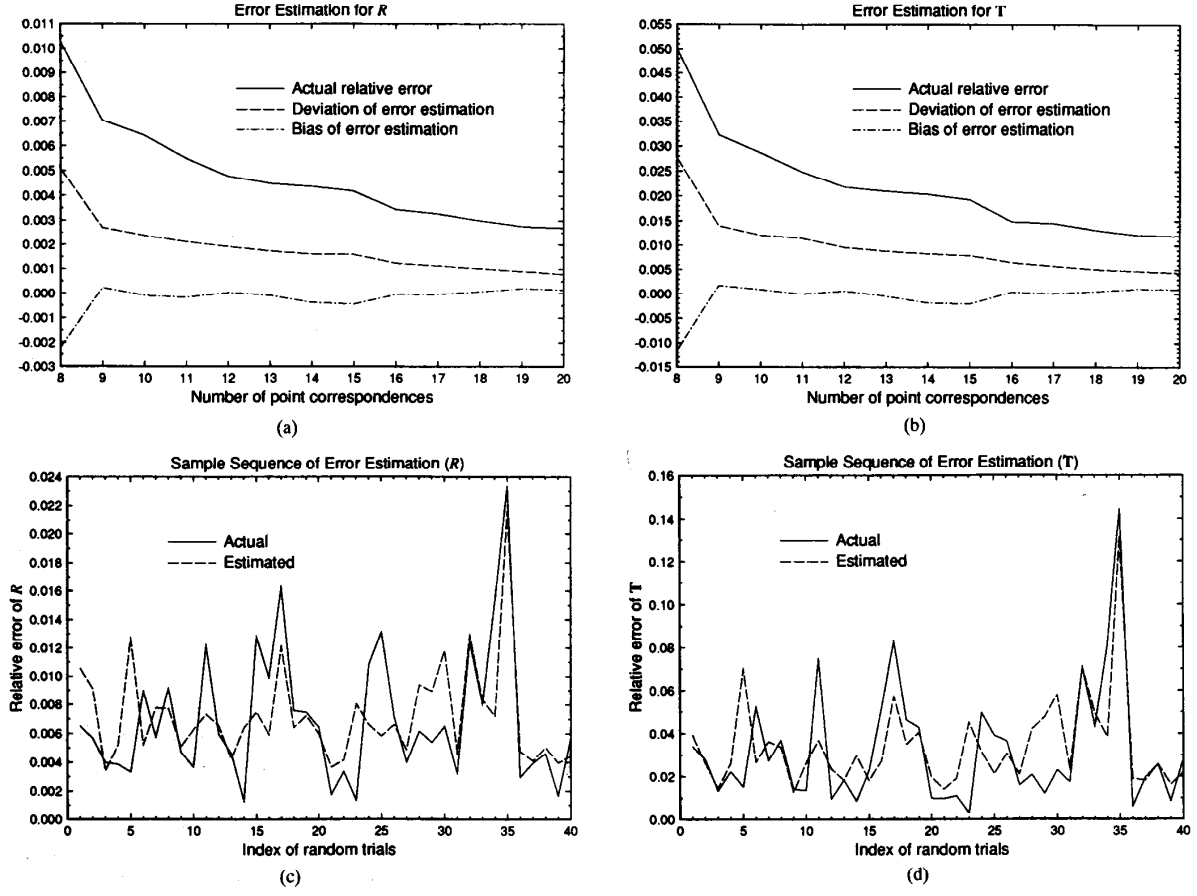


Fig. 14. Statistical record of error estimation. Actual relative error, deviation of error estimation, and bias of error estimation for (a) R and (b) T versus number of point correspondences. Sample sequences of estimated errors and actual errors with 9-point correspondences: (c) R and (d) T. Rotation axis: $(1, 0.9, 0.8)$; rotation angle: 5o; translation: $(0.5, -0.5, -3.0)$. 40 random trials.

(or image size is about 0.42 by 0.56 for the pin-hole camera model with a unit focal length shown in Fig. 1). The focal length of the camera is calibrated, but no nonlinear correction has been made for the camera. A two-view matcher computes an image displacement field and occlusion map on a pixel grid [53]. The displacement field gives point correspondences. Fig. 17 shows one of the two images of a scene in our laboratory (which is known as a Mac Scene). Samples of the displacement field computed are shown on a 13 by 14 grid in Fig. 17 and superimposed on the first image, which is extended to provide context for the peripheral areas of the image. Those $13 \times 14 = 182$ displacement vectors shown in Fig. 17 are used as point correspondences to compute motion parameters. The results are shown in Table IIa. Since no attempt was made to obtain the ground truth, we do not know the accuracy of those motion parameters. However, the image error is equal to about half a pixel width, as is shown in Table IIa, which seems to be satisfactory.

Assuming that the errors in the coordinates of the matching points are uncorrelated, the estimated variance of the errors is given by the squared image errors. The estimated errors of the computed motion parameters for the Mac Scene are shown in Table IIb. The results for other examples have been omitted due to space limitations.

VIII. Summary

This paper first discusses a type of motion with which the algorithms based on only the epipolar constraint are very sensitive to noise, especially under a small field of view. The analysis leads to the conclusion that it is important to use both components in the image coordinates of the points to determine the motion parameters in the presence of noise. The simulations showed that the use of both components (i.e., minimizing the image error) significantly improves the accuracy of the estimates compared with the use of only one component (i.e., minimizing the epipolar error), especially with a small field of view.

Both components of image coordinates and the constraints in the motion parameters are taken into account in a unified way by our approach to optimal estimation. The maximum likelihood estimator with independent Gaussian noise leads to minimization of the image error. With uncorrelated general noise distribution, minimizing the image error corresponds to minimum variance estimation for the locally linearized system.

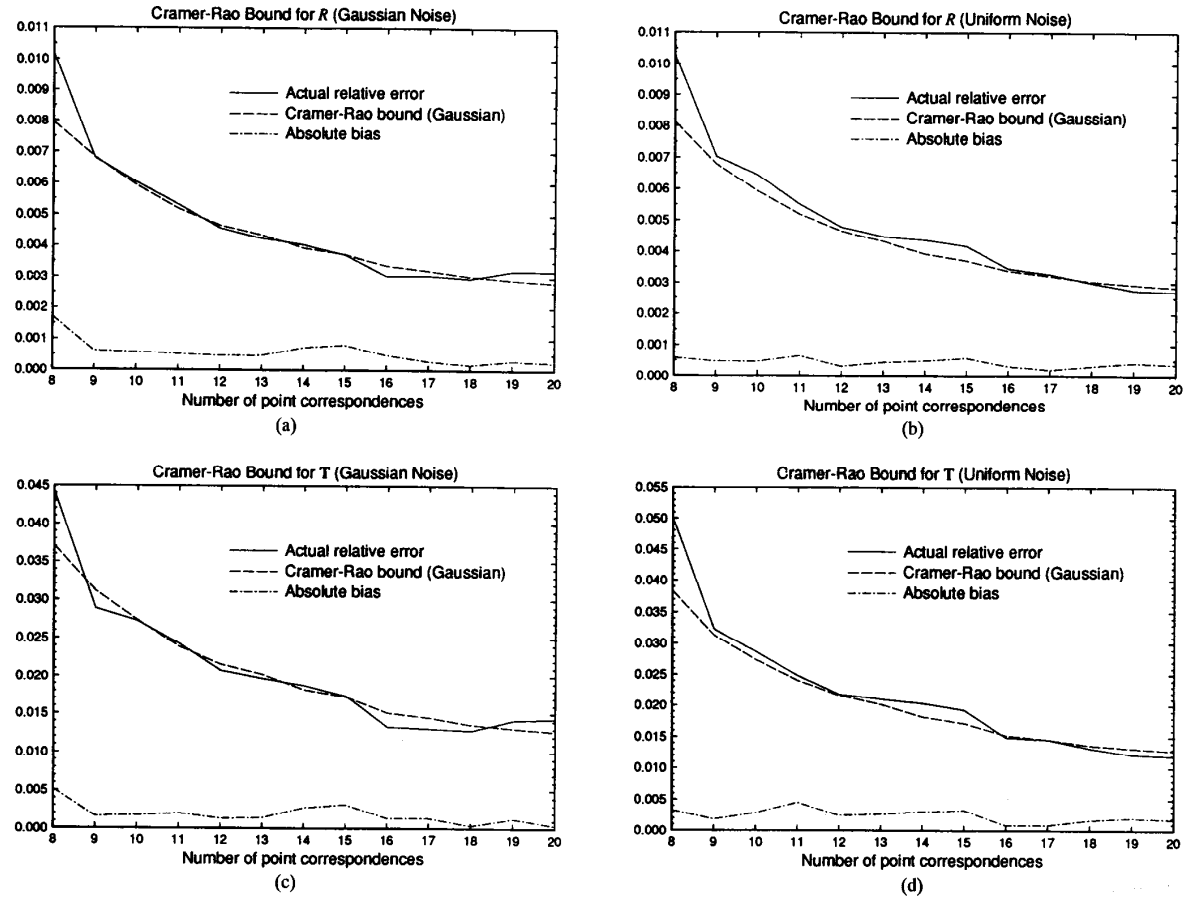


Fig. 15. Actual errors, the Cramér-Rao bound with Gaussian noise, and the absolute bias of the estimator versus the number of point correspondences. Comparison for R: (a) Gaussian noise added; (b) uniform noise added. Comparison for T: (c) Gaussian noise added; (d) uniform noise added. Rotation axis: $(1, 0.9, 0.8)$. Rotation angle: 5o. Translation: $(0.5, -0.5, -3.0)$. 40 random trials.

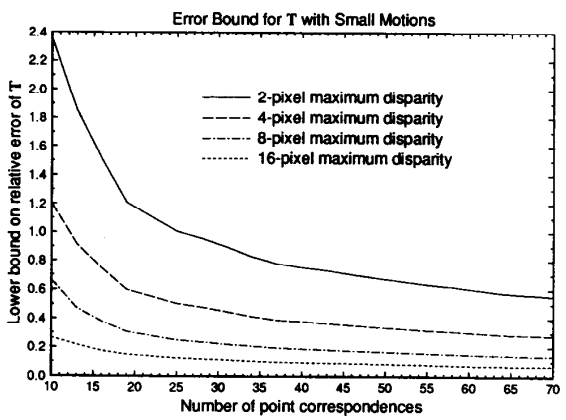


Fig. 16. Cramér-Rao bound for relative errors in translation under small motions. 10 random trials. Translation: $(k, k, 0)$. The value of k is such that the maximum disparity caused by translation is d pixels: $d = 2, 4, 8, 16$. Rotation axis: $(1, 0.9, 0.8)$; rotation angle: 5o.

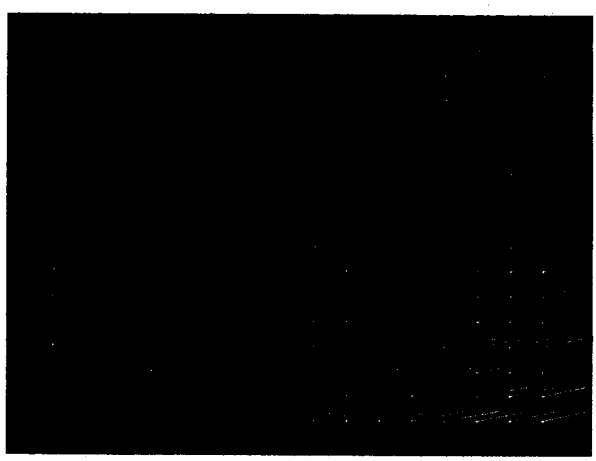


Fig. 17. One of the two views of a laboratory scene (Mac Scene) and samples of computed image plane displacement field.

Therefore, minimizing the image error is a good objective, even when the noise distribution is unknown.

The optimal estimation leads to a remarkable improvement over the preliminary estimates given by a linear algorithm.

TABLE IIa
DATA AND RESULTS FOR THE MAC SCENE

Parameters	x	y	z
Translation	0.016	0.991	0.13
Rotation axis	0.966	0.18	-0.19
Rotation angle		1.6°	
Image error		0.00 033	
Pixel width		0.00 094	

TABLE IIb
ESTIMATED ERRORS IN THE SOLUTION FOR THE MAC SCENE

Parameters	x	y	z
Errors of $\hat{T}$	0.0026	0.0011	0.012
Errors of Rotation axis	0.0091	0.021	0.023
Errors in rotation angle		0.14°	
Relative error in $\hat{T}$		0.012	
Relative error in rotation axis		0.032	

The stability of the motion estimation problem has been investigated in terms of theoretical limits. The Cramér-Rao lower error bound has been determined for the problem, and the experiments showed that the actual errors are quite close to the Cramér-Rao bound for unbiased estimators with Gaussian noise. This close-to-limit achievement for our nonlinear optimization problem is mainly due to the following:

1) The closed-form solution that provides a good initial guess
2) the optimal estimation that makes good use of all the available information and constraints
3) the effective batch processing.

A technique for estimating errors in the optimal estimates is introduced and tested. This provides a general framework of error estimation for iterative optimization algorithms.

The analysis and experiments on batch processing (the L-M or the Gauss-Newton method) and sequential processing (IEKF) lead to the following conclusions: For this highly nonlinear problem, the performance of the IEKF algorithm is inferior to that of the L-M algorithm in terms of accuracy, and the covariance matrices given by the IEKF may significantly underestimate the actual errors. To improve the computational efficiency for nonlinear problems with very long observation sequences, a sequential-batch processing approach [9] may be used.

The task of passive navigation or structure from motion has been recognized as problematic due to the instability observed. In this paper, the performance achieved for synthetic data and images of real-world scenes indicates that this task is fairly stable.

This is a journal version, for archival purposes, of our work on optimal motion estimation that we started in early 1986 [51] using an optimal objective function and the two-step approach, which was continued in 1987 [52], employing a maximum likehood method and extended in 1988 [54] to minimum variance estimation for unknown noise distribution. During the same early period in 1986, a technique of reconstruction and reprojection was developed by Toscani and Faugeras and appeared in the same workshop [44]. In 1988, Aisbett submitted her work on the iterative epipolar minimization [2], and around the same time, Spetsakis and Aloimonos published their work on optimal motion estimation using the epipolar minimization [43].

APPENDIX A

We need to derive (2.6). Equation (2.3) is equivalent to $\boldsymbol{T}_s^t(\boldsymbol{X}' \times R\boldsymbol{X}) = \boldsymbol{0}$. With additive noise $\boldsymbol{X}'(\epsilon) = \boldsymbol{X}' + \boldsymbol{\delta}_{\boldsymbol{X}}'$ and $\boldsymbol{X}(\epsilon) = \boldsymbol{X} + \boldsymbol{\delta}_{\boldsymbol{X}}$, we have

$$\begin{aligned}\boldsymbol{\eta} &\triangleq \boldsymbol{T}_s^t(\boldsymbol{X}'(\epsilon) \times R\boldsymbol{X}(\epsilon)) \\ &= \boldsymbol{T}_s^t\big((\boldsymbol{X}' + \boldsymbol{\delta}_{\boldsymbol{X}'}) \times R(\boldsymbol{X} + \boldsymbol{\delta}_{\boldsymbol{X}})\big) \\ &\simeq \boldsymbol{T}_s^t\big(\boldsymbol{X}' \times R\boldsymbol{X} + \boldsymbol{X}' \times R\boldsymbol{\delta}_{\boldsymbol{X}} + \boldsymbol{\delta}_{\boldsymbol{X}'} \times R\boldsymbol{X}\big)\end{aligned}$$

where the second-order term $\boldsymbol{\delta}_{\boldsymbol{X}'} \times (R\boldsymbol{\delta}_{\boldsymbol{X}})$ is neglected. Since $\boldsymbol{T}_s^t(\boldsymbol{X}' \times R\boldsymbol{X}) = \boldsymbol{0}$, we have

$$\begin{aligned}\boldsymbol{\eta} &\simeq \boldsymbol{T}_s^t(\boldsymbol{X}' \times R\boldsymbol{\delta}_{\boldsymbol{X}}) + \boldsymbol{T}_s^t(\boldsymbol{\delta}_{\boldsymbol{X}'} \times R\boldsymbol{X}) \\ &= \big(R^t(\boldsymbol{T}_s \times \boldsymbol{X}')\big)^t\boldsymbol{\delta}_{\boldsymbol{X}} - (\boldsymbol{T}_s \times R\boldsymbol{X})^t\boldsymbol{\delta}_{\boldsymbol{X}'}.\end{aligned}$$

The first two components of $\boldsymbol{X}$ are the image plane coordinates, and the last component is exactly 1. Assume that the first two components of its error vector $\boldsymbol{\delta}_{\boldsymbol{X}}$ are uncorrelated random noise with zero mean and variance σ^2, that is, $\mathbf{E}\boldsymbol{\delta}_{\boldsymbol{X}}\boldsymbol{\delta}_{\boldsymbol{X}}^t = \operatorname{diag}\{\sigma^2, \sigma^2, 0\}$. Similarly, assume $\boldsymbol{X}'$ has additive zero mean uncorrelated nose with variance σ^2. Then

$$\begin{aligned}\mathbf{E}\boldsymbol{\eta}\boldsymbol{\eta}^t &\simeq (R^t\boldsymbol{T}_s \times \boldsymbol{X}'^t))\operatorname{diag}\{\sigma^2, \sigma^2, 0\}R^t(\boldsymbol{T}_s \times \boldsymbol{X}') \\ &\quad + (\boldsymbol{T}_s \times R\boldsymbol{X})^t\operatorname{diag}\{\sigma^2, \sigma^2, 0\}(\boldsymbol{T}_s \times R\boldsymbol{X}) \\ &= \sigma^2\Big(||R^t(\boldsymbol{T}_s \times \boldsymbol{X}')||_{z=0}^2 + ||\boldsymbol{T}_s \times R\boldsymbol{X}||_{z=0}^2\Big)\end{aligned}$$

where $||(a, b, c)||_{z=0}^2 \triangleq a^2 + b^2$ is defined.

APPENDIX B

To prove the Cramér-Rao bound [8], [37] in (5.4), we first prove an inequality. Denoting $C_{\boldsymbol{XY}} \triangleq \mathbf{E}\boldsymbol{X}\boldsymbol{Y}^t$, then

$$C_{\boldsymbol{XX}} \geq C_{\boldsymbol{XY}}C_{\boldsymbol{YY}}^{\dagger}C_{\boldsymbol{XY}}^t. \tag{C.1}$$

In fact, for any matrix M, we have

$$\begin{aligned}\mathbf{o} &\leq \mathbf{E}(\boldsymbol{X} + M\boldsymbol{Y})(\boldsymbol{X} + M\boldsymbol{Y})^t \\ &= \mathbf{E}\big(\boldsymbol{X}\boldsymbol{X}^t + \boldsymbol{X}\boldsymbol{Y}^tM^t + M\boldsymbol{Y}\boldsymbol{X}^t + M\boldsymbol{Y}\boldsymbol{Y}^tM^t\big) \\ &= C_{\boldsymbol{XX}} + C_{\boldsymbol{XY}}M^t + MC_{\boldsymbol{XY}}^t + MC_{\boldsymbol{YY}}M^t.\end{aligned} \tag{C.2}$$

Letting $M = -C_{\boldsymbol{XY}}C_{\boldsymbol{YY}}^{\dagger}$ and using $C_{\boldsymbol{YY}}^{\dagger}C_{\boldsymbol{YY}}C_{\boldsymbol{YY}}^{\dagger} = C_{\boldsymbol{YY}}^{\dagger}$, (C.2) gives (C.1) immediately. The equality of (C.1) holds if and only of $\boldsymbol{X} + M\boldsymbol{Y} = \boldsymbol{0}$ almost everywhere.

Let $\boldsymbol{X} = \hat{\boldsymbol{m}} - \boldsymbol{b}(\boldsymbol{m})$ and $\boldsymbol{Y}^t = \boldsymbol{y}^t = \frac{\partial \ln p(\boldsymbol{u}, \boldsymbol{m})}{\partial \boldsymbol{m}}$. Using (C.1), we need to establish

$$\mathbf{E}\boldsymbol{X}\boldsymbol{Y}^t \triangleq C_{\boldsymbol{XY}} = B$$

to finish the proof. In fact

$$\begin{aligned}\mathbf{E}\boldsymbol{y}^t &= \mathbf{E}\frac{\partial \ln p}{\partial \boldsymbol{m}} = \int \frac{\partial \ln p}{\partial \boldsymbol{m}} p d\boldsymbol{u} = \int \frac{\partial p}{\partial \boldsymbol{m}} d\boldsymbol{u} \\ &= \frac{\partial}{\partial \boldsymbol{m}}\int p d\boldsymbol{u} = \frac{\partial}{\partial \boldsymbol{m}}1 = \boldsymbol{0}\end{aligned} \tag{C.3}$$

where assuming the integration and the differentiation can exchange the order (similar for the following). In addition

$$B = \frac{\partial b(\boldsymbol{m})}{\partial \boldsymbol{m}} = \frac{\partial \mathbf{E}\hat{\boldsymbol{m}}}{\partial \boldsymbol{m}} = \int \hat{\boldsymbol{m}} \frac{\partial p}{\partial \boldsymbol{m}} d\boldsymbol{u}$$
$$= \int \hat{\boldsymbol{m}} \frac{\partial \ln p}{\partial \boldsymbol{m}} p d\boldsymbol{u} = \mathbf{E}\hat{\boldsymbol{m}} \frac{\partial \ln p}{\partial \boldsymbol{m}} = \mathbf{E}\hat{\boldsymbol{m}}\boldsymbol{y}^t. \quad \text{(C.4)}$$

From (C.3) and (C.4), we have

$$\mathbf{E}\boldsymbol{X}\boldsymbol{Y}^t = \mathbf{E}(\hat{\boldsymbol{m}} - \boldsymbol{b}(\boldsymbol{m}))\boldsymbol{y}^t = \mathbf{E}\hat{\boldsymbol{m}}\boldsymbol{y}^t - \mathbf{E}\boldsymbol{b}(\boldsymbol{m})\boldsymbol{y}^t$$
$$= B - \boldsymbol{b}(\boldsymbol{m})\mathbf{0}^t = B.$$

□

APPENDIX C

The following is a version of the Kalman filtering algorithm:

$$\text{Kalman gain} : G_k = P_{k,k-1}A_k^t\left(A_k P_{k,k-1} A_k^t + C_k\right)^{-1}$$

$$\text{current estimate} : \hat{\boldsymbol{m}}_k = \hat{\boldsymbol{m}}_{k,k-1} + G_k(\boldsymbol{y}_k - A_k\hat{\boldsymbol{m}}_{k,k-1})$$

$$\text{current covariance} : P_k = P_{k,k-1} - G_k A_k P_{k,k-1}.$$

$$\text{predicted estimate} : \hat{\boldsymbol{m}}_{k+1,k} = \Phi_k\hat{\boldsymbol{m}}_k,$$

$$\text{predicted covariance} : P_{k+1,k} = \Phi_k P_k \Phi_k^t + Q_k.$$

The above computations are done for $k = 0, 1, 2, \ldots n$, initialized by $\hat{\boldsymbol{m}}_{0,-1} = \boldsymbol{a}$ and $P_{0,-1} = \mathbf{E}(\hat{\boldsymbol{m}}_{0,-1} - \boldsymbol{a})(\hat{\boldsymbol{m}}_{0,-1} - \boldsymbol{a})^t$. If *a priori* information for $\boldsymbol{m}_0$ is available, $\boldsymbol{m}_0$ is viewed as a random vector, assuming that $\boldsymbol{m}_0$ is uncorrelated with $\{\boldsymbol{\eta}_k\}$ and $\{\epsilon_k\}$. Then, $\boldsymbol{a}$ should be the expected $\boldsymbol{m}_0$: $\boldsymbol{a} = \mathbf{E}\boldsymbol{m}_0$. When there is no *a priori* information about $\boldsymbol{m}_0$, one can let $\boldsymbol{a} = \mathbf{0}$ and $P_{0,-1} = \infty I$. If $\boldsymbol{y}_k$ is a p-D vector, the update is called a rank-p update.

REFERENCES

[1] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects," *IEEE Trans. Patt. Anal. Machine Intell.* vol. PAMI-7, pp. 348–401, 1985.

[2] J. Aisbett, "An iterated estimation of the motion parameters of a rigid body from noisy displacement vectors," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 12, no. 11, pp. 1092–1098, 1990.

[3] B. D. Anderson and J. B. Moore, *Optimal Filtering*. Englewood Cliffs, NJ: Prentice-Hall, 1979.

[4] H. H. Baker, "Multiple-image computer vision," in *Proc. 41st Photogrammetric Week* (Stuttgart, West Germany), Sept. 1987, pp. 7–19.

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

[6] K. M . Brown and J. E. Dennis, "Derivative free analogues of the Levenberg-Marquardt and Gauss algorithms for nonlinear least squares approximation," *Numeriche Mathematik*, vol. 18, pp. 289–297, 1972.

[7] A. R. Bruss and B. K. Horn, "Passive navigation," *Comput. Vision Graphics Image Processing*, vol. 21, pp. 3–20, 1983.

[8] H. Cramér, *Mathematical Methods of Statistics*. Princeton, NJ: Princeton Univ. Press, 1946.

[9] N. Cui, J. Weng, and P. Cohen, "Extended structure and motion analysis from monocular image sequences," in *Proc. Third Int. Conf. Comput. Vision* (Osaka, Japan), Dec. 1990, pp. 222–229.

[10] O. D. Faugeras, N. Ayache, B. Faverjon, and F. Lustman, "Building visual maps by combining noisy stereo measurements," in *Proc. IEEE Int. Conf. Robotics Automat.* (San Francisco, CA), Apr. 1986, pp. 1433–1438.

[11] O. D. Faugeras, F. Lustman, and G. Toscani, "Motion and structure from point and line matches," in *Proc. Int. Conf. Comput. Vision* (London, England), June, 1987.

[12] R. J. Fitzgerald, "Divergence of the Kalman filter," *IEEE Trans. Automat. Contr.*, vol. AC-16, pp. 736–747, Dec. 1971.

[13] A. Gelb (Ed.), *Applied Optimal Estimation*. Cambridge, MA: MIT Press, 1974.

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

[15] R. Haralick and J. Lee, "The facet approach to optical flow," in *Proc. Image Understanding Workshop* (Arlington, VA), June 1983.

[16] D. Heeger, "Optical flow from spatiotemporal filters," in *Proc. First Int. Conf. Comput. Vision* (London, England), June 1987. pp. 181–190.

[17] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185–204, 1981.

[18] T. S. Huang and O. D. Faugeras, "Some properties of the E matrix in two-view motion estimation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11. no. 12, pp. 1310–1312, 1989.

[19] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.

[20] R. E. Kalman, *A New Approach to Linear Filtering and Prediction Problems, J. Basic Eng.*, 1960, pp. 35–45, Series 82D.

[21] J. K. Kearney, W. B. Thompson, and D. L. Boley, "Optical flow estimation: An error analysis of gradient-based methods with local optimization," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-9, no. 2, pp. 229–244, 1987.

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

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

[24] ——, "The reconstruction of a scene from two projections—Configurations that defeat the 8-point algorithm," in *Proc. 1st Conf. Artificial Intell. Applications* (Denver, CO), Dec. 5–7, 1984, pp. 395–397.

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

[26] ——, *Linear and Nonlinear Programming*. Reading, MA: Addison-Wesley, 1982, 2nd ed.

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

[28] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. New York: Academic, 1979, vol 1.

[29] ——, *Stochastic Models, Estimation, and Control*. New York: Academic, 1982, vol. 2.

[30] A. Mitiche and J. K. Aggarwal, "A computational analysis of time-varying images," *Handbook of Pattern Recognition and Image Processing* (T. Y. Young and K. S. Fu, Eds.). New York: Academic, 1986.

[31] L. Matthies, R. Szeliski, and T. Kanade, "Kalman filter-based algorithm for estimating depth from image sequences," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.* (Ann Arbor, MI), June 5–9, 1988, pp. 199–213.

[32] E. H. Moore, *General Analysis*, 1935, Memoirs, Amer. Philosoph. Soc. 1.

[33] H. -H. Nagel and W. Enkelmann, "An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 5, pp. 565–593, 1986.

[34] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*. New York: Academic, 1970.

[35] R. Penrose, "A generalized inverse for matrices," *Cambridge Philosoph. Soc.*, vol. 51, pp. 406–413, 1955.

[36] ——, "On best approximate solutions of linear matrix equations," *Cambridge Philosoph. Soc.*, vol. 52, pp. 17–19, 1956.

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

[38] P. Rives, E. Breuil, and B. Espiau, "Recursive estimation of 3D features using optical flow and camera motion," in *Proc. Conf. Intell. Autonomous Syst.*. Amsterdam: Elsevier, Dec. 1986.

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

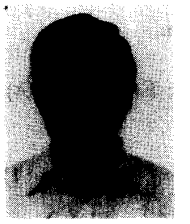
[40] F. H. Schlee, C. J. Standish, and N. F. Tota, "Divergence in the Kalman Filter," *AIAA J.*, vol. 5, pp. 1114–1120, June 1967.

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

[42] H. W. Sorenson (Ed), *Kalman Filtering: Theory and Application*. New York: IEEE Press, 1985.
[43] M. E. Spetsakis and J. Aloimonos, "Optimal motion estimation," in *Proc. IEEE Workshop Visual Motion*, Mar. 1989, pp. 229–237.
[44] G. Toscani and O. D. Faugeras, "Structure from motion using the reconstruction and reprojection technique," *Proc. IEEE Workshop Comput. Vision* (Miami, FL), Nov. 1987, pp. 345–348.
[45] H. L. Van Trees, *Detection, Estimation, and Modulation Theory*. New York: Wiley, 1969, vol. 1.
[46] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of 3-D motion parameters of rigid bodies with curved surfaces," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 6, no. 1, pp. 13–27, 1984
[47] A. Verri and T. Poggio, "Against quantitative optical flow," in *Proc. First Int. Conf. Comput. Vision* (London, England), June 1987. pp. 171–180.
[48] A. M. Waxman, B. Kamgar-Parsi, and M. Subbarao, "Closed-form solutions to image flow equations," in *Proc. First Int. Conf. Comput. Vision* (London, England), June 1987, pp. 12–24.
[49] J. Weng, N. Ahuja, and T. S. Huang, "Error analysis of motion parameters estimation from image sequences," in *Proc. Int. Conf. Comput. Vision* (London, England), June, 1987.
[50] J. Weng, T. S. 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.
[51] ——, "A two-step approach to optimal motion and structure estimation," in *Proc. IEEE Workshop Computer Vision* (Miami, FL), Nov. 1987, pp. 355–357.
[52] 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. Comput. Vision Patt. Recogn.* (Ann Arbor, MI), June 1988, pp. 381–386.
[53] ——, "Matching two perspective views," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 14, no. 8, pp. 806–825, Aug. 1992.
[54] ——, "Optimal motion and structure estimation," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.* (San Diego, CA), June 1989, pp. 144–152.
[55] J. Weng, T. S. Huang, and N. Ahuja, "Motion and structure from two perspective views: algorithms, error analysis and error estimation," in *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 5, pp. 451–476, 1989.
[56] S. S. Wilks, *Mathematical Statistics*. New York: Wiley, 1962.
[57] B. L. Yen and T. S. Huang, "Determining 3-D motion and structure of a rigid body using the spherical projection," *Comput. Vision Graphics Image Processing*, vol. 21, pp. 21–32, 1983.
[58] S. Zacks, *The Theory of Statistical Inference*. New York: Wiley, 1971.
[59] X. Zhuang, T. S. Huang, and R. M. Haralick, "Two-view motion analysis: A unified algorithm," *J. Opt. Soc. Amer.*, vol. 3, no. 9, pp. 1492–1500, Sept. 1986, ser. A.
[60] ——, "A simplified linear optic flow-motion algorithm," *Computer Vision Graphics Image Processing*, vol. 42, pp. 334–344, 1988.

Juyang Weng (M'88) received the B. S. degree from Fudan University, Shanghai, China, in 1982 and the M.S. and Ph.D. degrees from the University of Illinois, Urbana-Champaign, in 1985 and 1988, respectively, all in computer science.

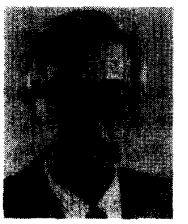
From September 1984 to December 1988, he was a research assistant at the Coordinated Science Laboratory, University of Illinois, Urbana-Champaign. In the summer of 1987, he was employed at IBM Los Angeles Scientific Center, Los Angeles, CA. Since January 1989, he has been a researcher at Centre de Recherche Informatique de Montréal, Montréal, Canada, while adjunctively with Ecole Polytechnique de Montréal. From October 1990 to August 1992, he was with the University of Illinois, Urbana-Champaign. Currently, he is an assistant professor with the Department of Computer Science, Michigan State University, East Lansing. His current research interests include computer vision, image processing, neural networks, object modeling and representation, parallel architecture for image processing, autonomous navigation, and artificial intelligence.



Narendra Ahuja (F'92) received the B.E. degree with honors in electrical engineering from the Birla Institute of Technology and Science, Pilani, India, in 1972, the M.E. degree with distinction in electrical communication engineering from the Indian Institute of Science, Bangalore, India, in 1974, and the Ph.D. degree in computer science from the University of Maryland, College Park, in 1979.

From 1974 to 1975, he was Scientific Officer in the Department of Electronics, Government of India, New Delhi. From 1975 to 1979, he was at the Computer Vision Laboratory, University of Maryland. Since 1979, he has been with the University of Illinois at Urbana-Champaign, where, since 1988, he has been a Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute. His interests are in computer vision, robotics, image processing, and parallel algorithms. He has been involved in teaching, research, consulting, and organizing conferences in these areas. His current research emphasizes integrated use of multiple image sources of scene information to construct 3-D descriptions of scenes, the use of acquired 3-D information for object manipulation and navigation, and multiprocessor architectures for computer vision.

Dr. Ahuja was selected as a Beckman Associate in the University of Illinois Center for Advanced Study for the years 1990 and 1991. He received the University Scholar Award (1985), hte Presidential Young Investigator Award (1984), the National Scholarship (1967–1972), and the President's Merit Award (1966). He has cooauthored the book *Pattern Models* (Wiley, 1983) with B. Schachter. He is an Associate Editor with the journals IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Computer Vision, Graphics, and Image Processing*, and *Journal of Mathematical Imaging and Vision*. He is a member of the American Association for Artificial Intelligence, the Society of Photo-Optical Instrumentation Engineers, and the Association for Computing Machinery.



Thomas S. Huang (F'79) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology (MIT), Cambridge.

He was with the Faculty of the Department of Electrical Engineering at MIT from 1963 to 1973 and with the Faculty of the School of Electrical Engineering and Director of the Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois, Urbana-Champaign, where he is currently Professor of Electrical and Computer Engineering and Research Professor at the Beckman Institute and the Coordinated Science Laboratory. During his sabbatical leaves, he has worked at the MIT Lincloln Laboratory, the IBM T. J. Watson Research Center, and the Rheinishes Lande Museum, Bonn, Germany, and held Visiting Professor positions at the Swiss Institute of Technology, Zurich and Lausanne, the University of Hannover, Germany, and INRS-Telecommunications of the University of Quebec, Montréal, Canada. He has served as a consultant to numerous industrial firms and government agencies both in the United States and abroad. His professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals. He has published 10 books and over 200 papers on network theory, digital filtering, image processing, and computer vision.

Dr. Huang is a Fellow of the Optical Society of America. He received a Guggenheim Fellowship (1971–1972), the A. V. Humboldt Foundation Senior U. S. Scientist Award (1976–1977), a Fellowship from the Japan Society for the Promotion of Science (1986), the IEEE Acoustics, Speech, and Signal Processing Society Technical Achievement Award (1987), and the University Scholar Award, University of Illinois at Urbana-Champaign (1990). He is an editor of the international journal *Computer Vision, Graphics, and Image Processing*, Editor of the *Springer Series in Information Sciences* (Springer-Verlag), and Editor of the *Research Annual Series on Advances in Computer Vision and Image Processing* (JAI Press).