Motion and Structure from Two Perspective Views: Algorithms, Error Analysis, and Error Estimation

JUYANG WENG, STUDENT MEMBER, IEEE, THOMAS S. HUANG, FELLOW, IEEE, AND NARENDRA AHUJA, SENIOR MEMBER, IEEE

Abstract—This paper deals with estimating motion parameters and the structure of the scene from point (or feature) correspondences between two perspective views.

First, a new algorithm is presented that gives a closed-form solution for motion parameters and the structure of the scene. The algorithm exploits redundancy in the data to obtain more reliable estimates in the presence of noise.

Then, an approach is introduced to estimating the errors in the motion parameters computed by the algorithm. Specifically, standard deviation of the error is estimated in terms of the variance of the errors in the image coordinates of the corresponding points. The estimated errors indicate the reliability of the solution as well as any degeneracy or near degeneracy that causes the failure of the motion estimation algorithm. The presented approach to error estimation applies to a wide variety of problems that involve least-squares optimization or pseudoinverse.

Finally, the relationships between errors and the parameters of motion and imaging system are analyzed. The results of the analysis show, among other things, that the errors are very sensitive to the translation direction and the range of field of view.

Simulations are conducted to demonstrate the performance of the algorithms, error estimation, as well as the relationships between the errors and the parameters of motion and imaging systems. The algorithms are tested on images of real world scenes with point correspondences computed automatically.

Index Terms—Computer vision, error analysis, error estimation, image sequence analysis, motion estimation, perturbation theory.

I. Introduction

One of our approaches to estimating three-dimensional motion parameters from image sequences can be divided into three steps. The first step is to establish feature correspondences for all pairs of consecutive image frames in a sequence. The second step is to estimate motion parameters for each such pair (called two-view motion parameters). The third step is using the two-view motion parameters to understand the local motion (short term motion, over, say, ten, twenty or more image frames) based on a model of object dynamics [30]. The approach is characterized by first estimating two-view motion parameters and then combining these parameters from more images. One of the major advantages of this approach is that features used for analysis does not have to be reliably traced over entire image sequence. In other words, a different set of features can be used for different pairs of images. Such an advantage is significant since a feature may disappear due to occlusion, or eventually leave the field of view as motion continues.

Alternatively, more image frames may be treated as a whole for correspondence based motion analysis. Such an approach requires that the features be identified and traced over all the image frames involved. For example, the unique closed-form solution can be derived, from three-frame line correspondences, for the motion parameters between every pair of the three images [32]. With correspondences through four or more image frames, fewer features are necessary than in the case of two-view motion analysis, provided that the motion over the entire image sequence is constant [26] or satisfies a known model [5].

This paper deals with two-view motion estimation. The results presented in this paper, however, can be extended to the approaches that require correspondences through three or more frames. The approach to analyzing and estimating errors is applicable to a wide class of problems that involve least-squares solution, minimum norm solution or pseudoinverse.

Estimating two-view motion parameters and structure of the scene conventionally involves two steps. The first step is establishing correspondences. The correspondences have been commonly obtained either from continuous approaches or discrete approaches. A continuous approach allows only a small interframe motion and computes optical flow fields [16], [24], [18], [29], [1], [15]. A discrete approach allows a relatively large interframe motion. Points (or corners and center of regions) [3], [8], [19], [37], [26], edges (or lines) [22], [6], [13], [10], [32], contours [7], and local intensity patterns [12], [17] can be utilized as features. Correspondences between features may be established through matching or interframe tracking. Recently, we have developed a two-view/stereo matcher that computes displacement fields from two images [33]. Information in pixel intensity, edges, shape of iso-intensity contours, intraregional smoothness and field discontinuity is employed in an integrated way to compute displacement fields and occlusion maps. The matcher has the advantages of both continuous approaches and discrete approach. For example, it allows an interframe motion which is relatively large compared

Manuscript received January 8, 1987; revised August 2, 1988. Recommended for acceptance by W. B. Thompson. This work was supported by the National Science Foundation under Grants IRI-86-05400 and ECS-83-52408.

The authors are with the Coordinated Science Laboratory, University of Illinois, Urbana, IL 61801.

IEEE Log Number 8926694.

to the continuous approaches and compute displacement fields on a dense pixel grid. For the experiments on real images presented in this paper, we use the displacement fields given by the above two-view matcher as point correspondences.

The second step concerns estimation of motion parameters and the structure of the scene from correspondences. Roach and Aggarwal [25] and Mitiche and Aggarwal [23] propose algorithms that solve for motion parameters directly from nonlinear equations (so, the algorithms of this type are called nonlinear algorithms). Nonlinear equations generally have to be solved through iterative methods with an initial guess or through global search. Iterative methods may diverge or converge to local minima. Searching in the space of motion parameters is computationally expensive. Linear algorithms solve linear equations and give closed-form solutions. Such algorithms using point correspondences have been developed independently by Longuet-Higgins [20], and Tsai and Huang [28]. The main advantages of linear algorithms over nonlinear ones are that they are fast, and uniqueness of solution is guaranteed except in degenerate cases.

Longuet-Higgins [21] derives a necessary and sufficient condition on the spatial configurations that cause the failure of the existing linear algorithms. The failure means that the algorithm fails to find the unique solution for such degenerate spatial configurations. Zhuang, Huang, and Haralick [35] give another necessary and sufficient condition for such a degeneracy. However, the existing linear algorithms essentially consider noise-free source data. High sensitivity to noise is reported in [9], [28]. To handle noise, Yasumoto and Medioni [34] include the magnitude of the rotation vector as a term of the objective function which is to be minimized. It is observed that suppressing rotation leads to no or negligible improvements. A consequence of their approach is that the estimated motion parameters are biased towards nonrotational interpretations if the regularization factor is not zero. Further, their search for the global minimum of the objective function in motion parameter space is computationally expensive. For estimating motion parameters from optical flow, Bruss and Horn [38] followed by Adiv [2], propose a method by which the motion parameters are estimated such that the discrepancy between the measured flow and that predicted from the computed motion parameters is minimized. However, their approach still lead to nonlinear equations and no effective methods are proposed to solve them.

Because the solution of a linear algorithm is generally suboptimal due to quantization and other errors, they can be further improved through optimization. For example, we introduce maximal likelihood estimation for this problem in [31]. Though the computation of optimal estimates still uses iterative numerical methods, a closed-form solution to be discussed in this paper generally serves as a very good initial guess. Starting with such a good initial guess, a locally optimal solution is generally globally optimal. Robustness of the linear algorithm is crucial since a bad initial guess will not lead to a globally optimal solution [31]. On the other hand, if many point correspondences are available (e.g., from displacement field [33]) and the motion is of stable type (the types of stable motion will be discussed in this paper), the solution of a robust linear algorithm is very close to the optimal one. As the number of point correspondences increases, the degree of improvement decreases [31] and the solution of the linear algorithm itself could be accepted.

In the presence of noise, several problems need to be solved. First, how can the algorithms make good use of the redundancy in the data to combat noise? Second, the noise may make a degenerate configuration nondegenerate mathematically. How can we check for the case of degeneracy or near-degeneracy? More generally, how can we assess the reliability of the solutions? Third, how are the errors related to the motion and system parameters? Any relative motion between the camera and the scene that yields large errors in solution should be avoided or treated accordingly in applications. Design of imaging systems should use parameters that result in stable estimation. We address these problems in this paper.

We first give a new algorithm aimed at simplicity, and insensitivity to noise. The algorithm exploits redundancy in the available data to improve accuracy of the solution. For the algorithm presented, we estimate the errors in the computed motion parameters. Large estimated errors indicate a degenerate or nearly degenerate configuration. The errors are estimated in terms of the variance of errors in the image coordinates of image points. Finally, the relationships between the errors and the motion and system parameters are investigated through qualitative analysis and quantitative experiments.

The motion estimation algorithm is discussed in the next section. Section III deals with error estimation. Section IV analyzes the dependency of the errors on motion and system parameters. Section V presents the simulation results for the algorithm, error analysis and error estimation. Section VI summarizes the main results.

II. A Two-View Motion Algorithm for the Presence of Noise

The objective is to reliably estimate the parameters of the relative motion between a camera and a rigid scene and the structure of the points from the point correspondences.

We first present an overview of the algorithm. Intermediate motion parameters are introduced which is called ''essential parameters'' by Tsai and Huang [28]. The essential parameters are a 3 by 3 matrix E, defined in terms of motion parameters. A set of equations are established that relates image coordinates of the feature points and the elements of matrix E. Since those equations are linear and homogeneous in the elements of E, the essential parameters E can be determined up to a scale factor. Then we solve for motion parameters from the essential parameters. Finally the relative depth (depth scaled by the magnitude of translation) of each point is determined from motion parameters and the observed projections of the point.

The essential parameter matrix E has 8 degrees of freedom (E is determined up to a scale factor). Each point correspondence gives one linear equation for E. This is why we need at least 8 point correspondences to solve for E. The relative motion between a camera and a rigid scene has 6 degrees of freedom (3 for rotation and 3 for translation). As we will see soon, the magnitude of the translation cannot be determined with one image sensor. Therefore the motion parameters to be determined have 5 degrees of freedom. To determine unknowns with 5 degrees of freedom from the matrix E with 8 degrees of freedom, we have overdeterminations. Those overdeterminations are fully exploited in the following algorithm to combat noise. In addition, there are a series of steps in which signs have to be determined. Stable methods are presented for determining those signs. In determining relative depth of each point, we again have overdeterminations. A least-squares solution is obtained for relative depths. Finally, because of the noise in the observed image coordinates of the feature points, the structures reconstructed directly from those observed points are not exactly related by a rigid motion between two images. The 3-D structure is corrected based on rigidity constraint.

We first introduce some notation. Matrices are denoted by capital italics. Vectors are denoted by bold fonts either capital or small. A three-dimensional column vector is specified by $(s_1, s_2, s_3)^T$. A vector is sometimes regarded as a column matrix. So vector operations such as cross product ($\times$) and matrix operations such as matrix multiplication are applied to three-dimensional vectors. Matrix operations precede vector operations. $\mathbf{0}$ denotes a zero vector. For a matrix $A = [a_{ij}]$, $\|A\|$ denotes the Euclidean norm of the matrix, i.e., $\|[a_{ij}]\| = \sqrt{\Sigma_{ij} a_{ij}^2}$. We define a mapping $[\,\cdot\,]_\times$ from a three-dimensional vector to a 3 by 3 matrix:

$$[(x_1, x_2, x_3)^T]_\times = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix}. \qquad (2.1)$$

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

$$\mathbf{X} \times \mathbf{Y} = [\mathbf{X}]_\times \mathbf{Y}. \qquad (2.2)$$

Let the coordinate system be fixed on the camera with the origin coinciding with the projection center of the camera, and the Z axis coinciding with the optical axis and pointing forward (Fig. 1). Without loss of generality, we assume that the focal length is unity. Namely image plane distance is measured in the units of focal length. Thus the image plane is located at $z = 1$. Visible objects are always located in front of the camera, i.e., $z > 0$.

Consider a point P on the object which is visible at two time instants. The following notation is used for the spatial vectors and the image vectors.

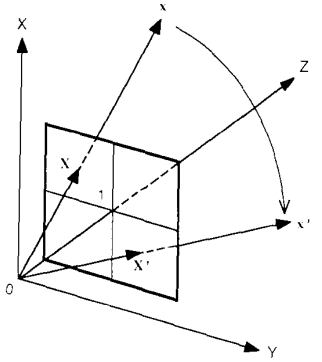


Fig. 1. Geometry and camera model of the setup.

$$\mathbf{x} = (x, y, z)^T$$

spatial vector of P at time t_1

$$\mathbf{x}' = (x', y', z')^T$$

spatial vector of P at time t_2

$$\mathbf{X} = (u, v, 1)^T = \left(\frac{x}{z}, \frac{y}{z}, 1\right)^T$$

image vector of P at time t_1

$$\mathbf{X}' = (u', v', 1)^T = \left(\frac{x'}{z'}, \frac{y'}{z'}, 1\right)^T$$

image vector of P at time t_2

where (u, v) and (u', v') are the image coordinates of the point. So, the spatial vector and image vector are related by

$$\mathbf{x} = z\mathbf{X}, \qquad \mathbf{x}' = z'\mathbf{X}'. \qquad (2.3)$$

Fig. 1 shows the geometry and the camera model of the setup.

Let R and $\mathbf{T}$ be the rotation matrix and the translational vector, respectively. The spatial points at the two time instants are related by

$$\mathbf{x}' = R\mathbf{x} + \mathbf{T} \qquad (2.4)$$

or for image vectors:

$$z'\mathbf{X}' = zR\mathbf{X} + \mathbf{T}. \qquad (2.5)$$

If $\|\mathbf{T}\| \neq 0$, from (2.5) we get

$$\frac{z'}{\|\mathbf{T}\|}\mathbf{X}' = R\frac{z}{\|\mathbf{T}\|}\mathbf{X} + \hat{\mathbf{T}} \qquad (2.6)$$

where

$$\hat{\mathbf{T}} = \frac{\mathbf{T}}{\|\mathbf{T}\|}. \qquad (2.7)$$

Given n corresponding image vector pairs at two time instants, $\mathbf{X}_i$ and $\mathbf{X}_i'$, $i = 1, 2, \cdots, n$, the algorithm solves for the rotation matrix R. If the translation vector $\mathbf{T}$ does not vanish, the algorithm solves for the translational direction represented by a unit vector $\hat{\mathbf{T}}$ and the relative depths $z_i / \| \boldsymbol{T} \|$ and $z_i' / \| \boldsymbol{T} \|$ for object points $\boldsymbol{x}_i$ and $\boldsymbol{x}_i'$, respectively. The magnitude of the translational vector ($\| \boldsymbol{T} \|$) and the absolute depths of the object points (z_i and z_i') cannot be determined by monocular vision. This is easy to see from (2.6), which still holds when $\| \boldsymbol{T} \|$, z_i, and z_i' are multiplied by any positive constant. In other words, multiplying the depths and $\| \boldsymbol{T} \|$ by the same scale factor does not change the images.

We shall first state the algorithm, and then justify each of the steps.

Algorithm

1) Solve for E:

Let $\boldsymbol{X}_i = (u_i, v_i, 1)^T$, $\boldsymbol{X}_i' = (u_i', v_i', 1)^T$, $i = 1, 2, \cdots, n$, be the corresponding image vectors of n ($n \geq 8$) points. Let

$$A = \begin{bmatrix} u_1u_1' & u_1v_1' & u_1 & v_1u_1' & v_1v_1' & v_1 & u_1' & v_1' & 1 \\ u_2u_2' & u_2v_2' & u_2 & v_2u_2' & v_2v_2' & v_2 & u_2' & v_2' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_nu_n' & u_nv_n' & u_n & v_nu_n' & v_nv_n' & v_n & u_n' & v_n' & 1 \end{bmatrix} \tag{2.8}$$

$$\boldsymbol{h} = (h_1, h_2, h_3, h_4, h_5, h_6, h_7, h_8, h_9)^T. \tag{2.9}$$

We solve for unit vector $\boldsymbol{h}$ such that

$$\| A\boldsymbol{h} \| = \min. \tag{2.10}$$

The solution of $\boldsymbol{h}$ is a unit eigenvector of A^TA associated with the smallest eigenvalue. Then E is determined by

$$E = [\boldsymbol{E}_1 \; \boldsymbol{E}_2 \; \boldsymbol{E}_3] = \sqrt{2} \begin{bmatrix} h_1 & h_4 & h_7 \\ h_2 & h_5 & h_8 \\ h_3 & h_6 & h_9 \end{bmatrix}. \tag{2.11}$$

2) Determine a Unit Vector $\boldsymbol{T}_s$ with $\hat{\boldsymbol{T}} = \pm \boldsymbol{T}_s$:

Solve for unit vector $\boldsymbol{T}_s$ such that

$$\| E^T \boldsymbol{T}_s \| = \min. \tag{2.12}$$

The solution of $\boldsymbol{T}_s$ is a unit eigenvector of EE^T associated with the smallest eigenvalue. If

$$\sum_i (\boldsymbol{T}_s \times \boldsymbol{X}_i') \cdot (E\boldsymbol{X}_i) < 0 \tag{2.13}$$

then $\boldsymbol{T}_s \leftarrow -\boldsymbol{T}_s$.

The summation in (2.13) is over several values of i's to suppress noise (usually three or four values of i will suffice).

3) Determine Rotation Matrix R:

Without noise we have

$$E = [\boldsymbol{T}_s]_\times R \tag{2.14}$$

or

$$R^T[-\boldsymbol{T}_s]_\times = E^T. \tag{2.15}$$

In the presence of noise, we find rotation matrix R such that

$$\left\| R^T[-\boldsymbol{T}]_\times - E^T \right\| = \min$$

$$\text{subject to: } R \text{ is a rotation matrix.} \tag{2.16}$$

Alternatively, we can find R directly. Let

$$W = [\boldsymbol{W}_1 \; \boldsymbol{W}_2 \; \boldsymbol{W}_3] = [\boldsymbol{E}_1 \times \boldsymbol{T}_s + \boldsymbol{E}_2 \times \boldsymbol{E}_3 \quad \boldsymbol{E}_2 \times \boldsymbol{T}_s + \boldsymbol{E}_3 \times \boldsymbol{E}_1 \quad \boldsymbol{E}_3 \times \boldsymbol{T}_s + \boldsymbol{E}_1 \times \boldsymbol{E}_2]. \tag{2.17}$$

Without noise we have $R = W$. In the presence of noise, we find rotation matrix R such that

$$\| R - W \| = \min \quad \text{subject to: } R \text{ is a rotation matrix.} \tag{2.18}$$

We can use either (2.16) or (2.18) to find R. They both have the form

$$\| RC - D \| = \min \quad \text{subject to: } R \text{ is a rotation matrix.} \tag{2.19}$$

Where $C = [\boldsymbol{C}_1 \; \boldsymbol{C}_2 \; \boldsymbol{C}_3]$, $D = [\boldsymbol{D}_1 \; \boldsymbol{D}_2 \; \boldsymbol{D}_3]$. The solution of (2.19) is as follows.

Define a 4 by 4 matrix B by

$$B = \sum_{i=1}^{3} B_i^T B_i \tag{2.20}$$

where

$$B_i = \begin{bmatrix} 0 & (\boldsymbol{C}_i - \boldsymbol{D}_i)^T \\ \boldsymbol{D}_i - \boldsymbol{C}_i & [\boldsymbol{D}_i + \boldsymbol{C}_i]_\times \end{bmatrix}. \tag{2.21}$$

Let $\boldsymbol{q} = (q_0, q_1, q_2, q_3)^T$ be a unit eigenvector of B associated with the smallest eigenvalue. The solution of rotation matrix R in (2.19) is

$$R = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_2q_1 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_3q_1 - q_0q_2) & 2(q_3q_2 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix}. \tag{2.22}$$

4) Check $\boldsymbol{T} = 0$. If $\boldsymbol{T} \neq 0$, Determine $\hat{\boldsymbol{T}} = \boldsymbol{T}_s$ or $\hat{\boldsymbol{T}} = -\boldsymbol{T}_s$:

Let α be a small threshold ($\alpha = 0$ without noise). If $\| \boldsymbol{X}_i' \times R\boldsymbol{X}_i \| / \| \boldsymbol{X}_i' \| \| \boldsymbol{X}_i \| \leq \alpha$ for all $1 \leq i \leq n$, then report $\boldsymbol{T} \approx 0$. Otherwise determine the sign for $\hat{\boldsymbol{T}}$: If

$$\sum_i (\boldsymbol{T}_s \times \boldsymbol{X}_i') \cdot (\boldsymbol{X}_i' \times R\boldsymbol{X}_i) > 0 \tag{2.23}$$

then $\hat{\boldsymbol{T}} = \boldsymbol{T}_s$. Otherwise $\hat{\boldsymbol{T}} = -\boldsymbol{T}_s$. Similar to (2.13), summation (2.23) is over several values of i.

5) If T Does Not Vanish, Estimate Relative Depths: For i, $1 \le i \le n$, find relative depth

$$\boldsymbol{Z}_i = \left(\frac{z_i'}{\|\boldsymbol{T}\|}, \frac{z_i}{\|\boldsymbol{T}\|}\right)^T = (\tilde{z}_i', \tilde{z}_i)^T \qquad (2.24)$$

such that

$$\left\|[\boldsymbol{X}_i' \; - R\boldsymbol{X}_i]\boldsymbol{Z}_i - \hat{\boldsymbol{T}}\right\| = \min \qquad (2.25)$$

using standard least-squares method for linear equations.

A simple method to correct structure based on rigidity constraint is as follows (see [31] for more robust methods). The corrected relative 3-D position (scaled by $\|\boldsymbol{T}\|^{-1}$) of point i at time t_2 equals to $\tilde{\boldsymbol{x}}_i' = (R(\tilde{z}_i\boldsymbol{X}_i) + \hat{\boldsymbol{T}} + \tilde{z}_i'\boldsymbol{X}_i')/2$. Its relative 3-D position (scaled by $\|\boldsymbol{T}\|^{-1}$) at time t_1 equals to $\tilde{\boldsymbol{x}}_i = R^{-1}(\tilde{\boldsymbol{x}}_i' - \hat{\boldsymbol{T}})$.

We now justify each step of the algorithm.

For Step 1): Let $\boldsymbol{T}_s$ be a unit vector that is aligned with $\boldsymbol{T}$, i.e.,

$$\boldsymbol{T}_s \times \boldsymbol{T} = 0. \qquad (2.26)$$

Precrossing both sides of (2.6) by $\boldsymbol{T}_s$ we get [using (2.1), (2.2)]:

$$\frac{z'}{\|\boldsymbol{T}\|}\boldsymbol{T}_s \times \boldsymbol{X}' = \frac{z}{\|\boldsymbol{T}\|}[\boldsymbol{T}_s]_\times R\boldsymbol{X}. \qquad (2.27)$$

Premultiplying both sides of (2.27) by $\boldsymbol{X}'^T$ (inner product of between vectors), we get:

$$\boldsymbol{X}'^T[\boldsymbol{T}_s]_\times R\boldsymbol{X} = 0 \qquad (2.28)$$

since $\boldsymbol{X}'^T(\boldsymbol{T}_s \times \boldsymbol{X}') = 0$ and $z > 0$. Define E to be

$$E = [\boldsymbol{T}_s]_\times R = [\boldsymbol{T}_s \times \boldsymbol{R}_1 \quad \boldsymbol{T}_s \times \boldsymbol{R}_2 \quad \boldsymbol{T}_s \times \boldsymbol{R}_3]$$
$$= [\boldsymbol{E}_1\, \boldsymbol{E}_2\, \boldsymbol{E}_3] \qquad (2.29)$$

where $R = [\boldsymbol{R}_1\, \boldsymbol{R}_2\, \boldsymbol{R}_3]$. From the definition of $\boldsymbol{T}_s$, the sign of E is arbitrary since the sign of $\boldsymbol{T}_s$ is arbitrary (as long as the sign of $\boldsymbol{T}_s$ and that of E match such that (2.29) holds). Using (2.29), the definition of E, we rewrite (2.28) as

$$\boldsymbol{X}'^T E \boldsymbol{X} = 0. \qquad (2.30)$$

Our objective is to find E from the image vectors $\boldsymbol{X}$ and $\boldsymbol{X}'$. Each point correspondence gives one equation (2.30) which is linear and homogeneous in the elements of E. n point correspondences give n such equations. Let

$$E = \begin{bmatrix} e_1 & e_4 & e_7 \\ e_2 & e_5 & e_8 \\ e_3 & e_6 & e_9 \end{bmatrix}, \qquad \boldsymbol{E} = (e_1\, e_2 \cdots e_9)^T. \qquad (2.31)$$

Rewriting (2.30) in the elements of E using n point correspondences, we have

$$A\boldsymbol{E} = 0. \qquad (2.32)$$

With noise we use (2.10). The solution of $\boldsymbol{h}$ in (2.32) is then equal to $\boldsymbol{E}$ up to a scale factor if $\operatorname{rank}(A) = 8$. The rank of the n by 9 matrix A cannot be larger than 8 since $\boldsymbol{E}$ is a nonzero solution of (2.32). Longuet-Higgins [21] gives a necessary and sufficient condition for the matrix A to have a rank less than 8.

Since the sign of E is arbitrary, we need only to find the Euclidean norm of E to determine E (equivalently $\boldsymbol{E}$) from $\boldsymbol{h}$. Let $\boldsymbol{T}_s = (s_1, s_2, s_3)^T$. Noticing $\boldsymbol{T}_s$ is a unit vector and using (2.29), we get

$$\begin{aligned} \|E\|^2 &= \text{trace}\,\{EE^T\} \\ &= \text{trace}\left\{[T_s]_\times R([T_s]_\times R)^T\right\} \\ &= \text{trace}\left\{[T_s]_\times([T_s]_\times)^T\right\} \end{aligned}$$

$$= \text{trace}\left\{\begin{bmatrix} 0 & -s_3 & s_2 \\ s_3 & 0 & -s_1 \\ -s_2 & s_1 & 0 \end{bmatrix}\begin{bmatrix} 0 & s_3 & -s_2 \\ -s_3 & 0 & s_1 \\ s_2 & -s_1 & 0 \end{bmatrix}\right\}$$

$$= 2(s_1^2 + s_2^2 + s_3^2) = 2.$$

So, $\boldsymbol{E} = \sqrt{2}\boldsymbol{h}$. This gives (2.11).

For Step 2): We determine $\boldsymbol{T}_s$. By (2.29), $\boldsymbol{T}_s$ is orthogonal to all three columns of E. We get $E^T\boldsymbol{T}_s = 0$. With noise, we use (2.12).

It is easy to prove that the rank of E is always equal to 2. In fact, let $\boldsymbol{Q}_2$ and $\boldsymbol{Q}_3$ be such that $Q = [\boldsymbol{T}_s\, \boldsymbol{Q}_2\, \boldsymbol{Q}_3]$ is an orthonormal 3 by 3 matrix. $S = R^TQ$ is also orthonormal. Postmultiplying the two sides of the first equation of (2.29) by S, we get

$$ES = [\boldsymbol{T}_s]_\times RS = [\boldsymbol{T}_s]_\times Q = [0 \quad \boldsymbol{T}_s \times \boldsymbol{Q}_2 \quad \boldsymbol{T}_s \times \boldsymbol{Q}_3].$$

We see the second and the third columns of ES are orthonormal from the definition of Q. So rank $\{E\}$ = rank $\{ES\} = 2$.

Since rank $\{E\} = 2$, the unit vector $\boldsymbol{T}_s$ is uniquely determined up to a sign by (2.12). To determine the sign of $\boldsymbol{T}_s$ such that (2.29) holds, we rewrite (2.27) using $E = [\boldsymbol{T}_s]_\times R$:

$$\frac{z'}{\|\boldsymbol{T}\|}\boldsymbol{T}_s \times \boldsymbol{X}' = \frac{z}{\|\boldsymbol{T}\|}E\boldsymbol{X}. \qquad (2.33)$$

Since $z > 0$ for all the visible points, from (2.33) we know the two vectors $\boldsymbol{T}_s \times \boldsymbol{X}_i'$ and $E\boldsymbol{X}_i$ have the same directions. If the sign of $\boldsymbol{T}_s$ is wrong, they have the opposite directions. So if (2.13) holds, the sign of $\boldsymbol{T}_s$ should be changed.

For Step 3): In steps 1) and 2) we found E and $\boldsymbol{T}_s$ that satisfy (2.29). R can be determined directly by (2.17). We now prove W in (2.17) is equal to R without noise:

$$R = [\boldsymbol{R}_1\, \boldsymbol{R}_2\, \boldsymbol{R}_3] = [\boldsymbol{E}_1 \times \boldsymbol{T}_s + \boldsymbol{E}_2 \times \boldsymbol{E}_3$$
$$\boldsymbol{E}_2 \times \boldsymbol{T}_s + \boldsymbol{E}_3 \times \boldsymbol{E}_1 \quad \boldsymbol{E}_3 \times \boldsymbol{T}_s + \boldsymbol{E}_1 \times \boldsymbol{E}_2].$$
$$(2.34)$$

Using the identity equation $(\boldsymbol{a} \times \boldsymbol{b}) \times \boldsymbol{c} = (\boldsymbol{a} \cdot \boldsymbol{c})\boldsymbol{b} - (\boldsymbol{b} \cdot \boldsymbol{c})\boldsymbol{a}$ and (2.29), we get

$$\begin{aligned} &\boldsymbol{E}_1 \times \boldsymbol{T}_s + \boldsymbol{E}_2 \times \boldsymbol{E}_3 \\ &\quad = (\boldsymbol{T}_s \times \boldsymbol{R}_1) \times \boldsymbol{T}_s + (\boldsymbol{T}_s \times \boldsymbol{R}_2) \times (\boldsymbol{T}_s \times \boldsymbol{R}_3) \\ &\quad = (\boldsymbol{T}_s \cdot \boldsymbol{T}_s)\boldsymbol{R}_1 - (\boldsymbol{R}_1 \cdot \boldsymbol{T}_s)\boldsymbol{T}_s \\ &\qquad + (\boldsymbol{T}_s \cdot (\boldsymbol{T}_s \times \boldsymbol{R}_3))\boldsymbol{R}_2 - (\boldsymbol{R}_2 \cdot (\boldsymbol{T}_s \times \boldsymbol{R}_3))\boldsymbol{T}_s \\ &\quad = \boldsymbol{R}_1 - (\boldsymbol{R}_1 \cdot \boldsymbol{T}_s)\boldsymbol{T}_s + (\boldsymbol{R}_2 \cdot (\boldsymbol{R}_3 \times \boldsymbol{T}_s))\boldsymbol{T}_s \\ &\quad = \boldsymbol{R}_1 - (\boldsymbol{R}_1 \cdot \boldsymbol{T}_s)\boldsymbol{T}_s + ((\boldsymbol{R}_2 \times \boldsymbol{R}_3) \cdot \boldsymbol{T}_s)\boldsymbol{T}_s \\ &\quad = \boldsymbol{R}_1 - (\boldsymbol{R}_1 \cdot \boldsymbol{T}_s)\boldsymbol{T}_s + (\boldsymbol{R}_1 \cdot \boldsymbol{T}_s)\boldsymbol{T}_s \\ &\quad = \boldsymbol{R}_1. \end{aligned}$$

This proves that the first column of R is correct. Similarly we can prove that the remaining columns of R are correct.

To solve the problem of (2.19), we represent the rotation matrix R in terms of a unit quaternion $\boldsymbol{q}$. R and $\boldsymbol{q}$ are related by (2.22) [4], [14]. We have (see Appendix B)

$$\|RC - D\|^2 = \boldsymbol{q}^t B \boldsymbol{q} \qquad (2.35)$$

where B is defined in (2.20) and (2.21). The solution of (2.19) is then reduced to the problem of minimization of a quadratic. The solution of the unit vector $\boldsymbol{q}$ in (2.35) is then a unit eigenvector of B associated with the smallest eigenvalue.

For Step 4): Precrossing both sides of (2.5) by $\boldsymbol{X}'$, we get

$$0 = z\boldsymbol{X}' \times R\boldsymbol{X} + \boldsymbol{X}' \times \boldsymbol{T}. \qquad (2.36)$$

If $\boldsymbol{T} = 0$, for any point $\boldsymbol{X}'$ we have (note $z > 0$)

$$\boldsymbol{X}' \times R\boldsymbol{X} = 0. \qquad (2.37)$$

If $\boldsymbol{T} \neq 0$, $\boldsymbol{X}' \times \boldsymbol{T} \neq 0$ holds for all the points $\boldsymbol{X}'$ (except at most one). So (2.37) cannot hold for all points from (2.36). In the algorithm, we normalize the image vectors in (2.37) and give a tolerance threshold α in the presence of noise.

From (2.36), if $\hat{\boldsymbol{T}} = \boldsymbol{T}_s$, then $\boldsymbol{T}_s \times \boldsymbol{X}'$ and $\boldsymbol{X}' \times R\boldsymbol{X}$ have the same directions. Otherwise, they have the opposite directions since $\hat{\boldsymbol{T}} = -\boldsymbol{T}_s$. We use the sign of the inner product of the two vectors in (2.23) to determine the sign of $\hat{\boldsymbol{T}}$.

For Step 5): The equations for the least-squares solution (2.24) are directly from (2.6). The idea for correcting structure based on rigidity is as follows. Moving the recovered 3-D point at time t_1 using the estimated rotation and translation, its new position should be exactly at the recovered position at time t_2, if data are noise free. However, in the presence of noise, they are not exactly equal. Here we choose a simple method: the midpoint between those two positions are chosen as the corrected solution for the position of the point at time t_2. Moving the midpoint back gives the corrected 3-D position of the point at time t_1. A more detailed discussion about correcting structure can be found in [31], where noise distribution is considered to obtain a more robust estimate.

In summary, we have derived a close-form solution of the problem. Given 8 or more point correspondences, the algorithm first solves for the essential parameter matrix E. Then the motion parameters are obtained from E. Finally the spatial structure is derived from the motion parameters. All the steps of the algorithm use the redundancy in the data to combat noise. As the results of determining the signs in (2.13) and (2.23), the computations of three false solutions required by other existing linear algorithms [20], [28], [35] are avoided. These steps for determining signs are stable in the presence of noise, since the decisions are made based on the signs of the inner product of the two vectors which are in the same or opposite direction without noise. Summations over several points in (2.13) and (2.23) suppress the cases where two noise-corrupted small vectors happen to be used, whose inner products are close to zero and the signs are unreliable.

If $\boldsymbol{T} \neq 0$ and the spatial configuration is nondegenerate, the rank of A is 8. In this case, we can determine the unit vector $\boldsymbol{h}$ in (2.10) up to a sign. If $\boldsymbol{T} = 0$ any unit vector $\boldsymbol{T}_s$ satisfies (2.27) and so E, and correspondingly, the unit vector $\boldsymbol{h}$ have two degrees of freedom (notice $\boldsymbol{T}_s$ and $\boldsymbol{h}$ are restricted to be unit vectors). Therefore, A in (2.8) has a rank less than or equal to 6. If $\boldsymbol{T} = 0$, relative depths of the points cannot be determined. However, the rotation parameters are still determined even if $\boldsymbol{T} = 0$.

The next section discusses how to estimate the reliability of the computed E and motion parameters.

III. Error Estimation

One way to do error analysis is determining worst case bounds on errors. Such bounds are useful for applications with small errors such as computer round off errors. Since computer word length is large enough for most applications, the worst case bounds are generally tolerable. However, in problems where redundancy is utilized to combat noise and the errors in the data are not very small, the conventional worst case analysis usually renders an overly conservative bound. This can be visualized by considering the upper bound of a random variable with Gaussian distribution. Since the bound is almost never reached, the utility of the bound is very limited. In many problems with redundant data, however, the error level of the solution is relatively stable for a fixed level of input noise. This stability is due to the redundancy in observations. For example, consider a Gaussian random variable with a small variance, denoting error in solutions. Motion analysis from images is one of such examples. A conventional worst case error bound analysis gives a value that is often larger than the true value itself. If the least-squares solution is derived from a large amount of data, the variance of the error distribution of the solution is small. This make it possible for us to estimate errors in the solutions. In this section we investigate how to estimate errors in the solution instead of deriving a worst case bound which is very large and almost never reached. The approach discussed in this section is applicable to problems where least-squares solution, minimum-norm solution or pseudoinverse is involved, since they essentially reduce to an eigenvalue and eigenvector problem.

The sources of errors in the image coordinates include spatial quantization, feature detector errors, point mismatching and camera distortion. In a system that is well calibrated so that systematic errors are negligible, errors in the image coordinates of a feature can be modeled by random variables. These errors result in the errors of the estimates of the motion parameters. Some spatial configurations of the points are relatively insensitive to the errors in the image coordinates of the points, but some are very sensitive. For example, if a spatial configuration of the points is degenerate mathematically but the errors in the measured image coordinates make them nondegenerate, any estimates under such configuration is useless. If we move a single point slightly, so that the configuration stops being degenerate, such a configuration must be very sensitive to noise.

Formally, let the image coordinates of all the points be denoted by I, and the errors in the image coordinates of these points be denoted by a random variable ϵ. The error e in the estimated motion parameters is a function of I and ϵ. Denoting this function by f, informally we can write:

$$e = f(I, \epsilon). \tag{3.1}$$

Our goal is to estimate the error e given the images I. However we do not know ϵ. If we can estimate the standard deviation of e (with ϵ as a random variable) given the noise-corrupted image I, we can use it to estimate the errors of the estimates. The images I corresponding to a degenerate or nearly degenerate spatial configuration should yield large estimates of e and that corresponding to a stable configuration should yield small estimates.

For the following discussion, we assume that the noise in the image coordinates has zero mean and known variance. For example, the spatial quantization noise can be well modeled by a uniform distribution with the range corresponding to the width of the pixels. The variance of the feature detector error can also be estimated empirically. We also assume the noises at the different points are uncorrelated, and the noises in the two components of the image coordinates are uncorrelated. This assumption of uncorrelatedness is not exactly true in reality. However the correlation between them can be regarded negligible (in the first order perturbation). We estimate the standard deviation of the errors in the motion parameters on the basis of first order perturbation, in other words, we estimate the "linear terms" of the errors.

For conciseness, we use the following notation: I_m denotes an m by m identity matrix. A matrix A without noise is denoted by A itself and its elements denoted by the corresponding small letters a_{ij}, i.e., $A = [a_{ij}]$. The noise matrix of A with the same size is denoted by Δ_A. The noise-corrupted A is denoted by $A(\epsilon)$. We have

$$A(\epsilon) = A + \Delta_A. \tag{3.2}$$

Similarly for vectors, we use δ with corresponding subscript to denote the noise vectors:

$$\boldsymbol{X}(\epsilon) = \boldsymbol{X} + \delta_x. \tag{3.3}$$

Γ with the corresponding subscript is used to denote the covariance matrix of the noise vector (considering only the first order errors, the means of the errors are zero):

$$\Gamma_x = \mathrm{E}\{\delta_x \delta_x^T\} \tag{3.4}$$

where E denotes expectation. A matrix $A = [\boldsymbol{A}_1 \, \boldsymbol{A}_2 \cdots \boldsymbol{A}_n]$ is associated with a corresponding vector $\boldsymbol{A}$ with

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{A}_1 \\ \boldsymbol{A}_2 \\ \vdots \\ \boldsymbol{A}_n \end{bmatrix}. \tag{3.5}$$

Similarly Γ_A denotes the corresponding covariance matrix of the vector $\boldsymbol{A}$ associated with matrix A. δ_A denotes the perturbation vector associated with the perturbation matrix Δ_A. "$\doteq$" is used in the equations to define new variables when the variable to be defined is obvious.

Assuming two variables a and b with small errors:

$$a(\epsilon) = a + \delta_a, \qquad b(\epsilon) = b + \delta_b \tag{3.6}$$

we have

$$a(\epsilon)\, b(\epsilon) = ab + \delta_a b + a\delta_b + \delta_a \delta_b \doteq ab + \delta_{ab}. \tag{3.7}$$

The error in $a(\epsilon)\, b(\epsilon)$ is

$$\delta_{ab} = \delta_a b + a\delta_b + \delta_a \delta_b \cong \delta_a b + a\delta_b. \tag{3.8}$$

In the last approximation we keep the linear terms of the error and ignore the higher order terms. Later in this paper we use the sign "$\cong$" for the equations that are equal in the linear terms ("$\approx$" for the approximate equality in the usual sense). Considering a small perturbation in the original data, we analyze the linear terms of the corresponding perturbation (first order perturbation) of the final results to estimate its error. In our problem, the noise or errors are from the image coordinates. The final results are the motion parameters calculated by the algorithm that is presented in the previous section.

The algorithm presented involves the calculation of the eigenvectors of a symmetrical matrix. With small perturbation in the matrix, we need to known the corresponding perturbation in its eigenvectors. We have the following theorem.

Theorem: Let $A = [a_{ij}]$ be an n by n symmetrical matrix and H be an orthonormal matrix such that

$$H^{-1}AH = \mathrm{diag}\,\{\lambda_1, \lambda_2, \cdots, \lambda_n\} \tag{3.9}$$

(where diag $\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$ denotes the diagonal matrix with the corresponding diagonal elements). Let the eigenvalues be ordered in nondecreasing order. Without loss of generality, consider the eigenvalue λ_1. Assuming λ_1 is a simple eigenvalue, we have

$$\lambda_1 < \lambda_2 \le \lambda_3 \le \cdots \le \lambda_n. \qquad (3.10)$$

Let

$$H = [\boldsymbol{h}_1 \; \boldsymbol{h}_2 \cdots \boldsymbol{h}_n]. \qquad (3.11)$$

Let X be an eigenvector of A associated with λ_1. $\boldsymbol{X}$ is then a vector in span $\{\boldsymbol{h}_1\}$ (the linear space spanned by $\boldsymbol{h}_1$). Let $X(\epsilon)$ be the eigenvector of the perturbed matrix $A(\epsilon) = A + \Delta_A$ associated with the perturbed eigenvalue $\lambda_1(\epsilon)$. $X(\epsilon)$ can be written as

$$X(\epsilon) = \boldsymbol{X} + \boldsymbol{\delta}_x \qquad (3.12)$$

with $\boldsymbol{\delta}_x \in$ span $\{\boldsymbol{h}_2, \boldsymbol{h}_3, \cdots, \boldsymbol{h}_n\}$. Letting ϵ be the maximum absolute value of the elements in $\Delta_A = [\delta_{a_{ij}}]$, we have

$$\Delta_A = \epsilon B \qquad (3.13)$$

where $B = [b_{ij}]$, with $b_{ij} = \delta_{a_{ij}}/\epsilon$. Therefore, $|b_{ij}| \le 1$, $1 \le i \le n$, $1 \le j \le n$. Then for sufficiently small ϵ, the perturbation of λ_1 can be expressed by a convergent series in ϵ:

$$\delta_{\lambda_1} \triangleq \lambda_1(\epsilon) - \lambda_1 = p_1\epsilon + p_2\epsilon^2 + p_3\epsilon^3 + \cdots \qquad (3.14)$$

and the perturbation vector $\boldsymbol{\delta}_x$ can be expressed by a convergent vector series in the space span $\{\boldsymbol{h}_2, \boldsymbol{h}_3, \cdots, \boldsymbol{h}_n\}$. In other words, letting $H_2 = [\boldsymbol{h}_2, \boldsymbol{h}_3, \cdots, \boldsymbol{h}_n]$, then for sufficiently small positive ϵ, there exist $(n-1)$-dimensional vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \cdots$ such that

$$\boldsymbol{\delta}_x = \epsilon H_2 \boldsymbol{g}_1 + \epsilon^2 H_2 \boldsymbol{g}_2 + \epsilon^3 H_2 \boldsymbol{g}_3 + \cdots. \qquad (3.15)$$

The linear term (in ϵ) in (3.14) is given by

$$p_1\epsilon = \boldsymbol{h}_1^T \Delta_A \boldsymbol{h}_1. \qquad (3.16)$$

The linear term (in ϵ) in (3.15) is given by

$$\epsilon H_2 \boldsymbol{g}_1 = H\Delta H^T \Delta_A \boldsymbol{X} \qquad (3.17)$$

where

$$\Delta = \text{diag}\left\{0, (\lambda_1 - \lambda_2)^{-1}, \cdots, (\lambda_1 - \lambda_n)^{-1}\right\}. \qquad (3.18)$$

That is, suppressing the second and higher order terms (i.e., considering first order perturbation), for the eigenvalue we have

$$\delta_{\lambda_1} \cong \boldsymbol{h}_1^T \Delta_A \boldsymbol{h}_1$$

and for the eigenvector:

$$\boldsymbol{\delta}_x \cong H\Delta H^T \Delta_A \boldsymbol{X}. \qquad (3.19)$$

Proof: See Appendix A.

The above theorem gives the first order perturbation of the eigenvector associated with a simple eigenvalue λ_1. A similar result holds for other simple eigenvalues. For example, to give the first order perturbation of the eigenvector $\boldsymbol{X}_2$ associated with a simple eigenvalue λ_2, we just need to modify the matrix Δ in $\boldsymbol{\delta}_{x_2} \cong H\Delta H^t \Delta_A \boldsymbol{X}_2$:

$$\Delta = \text{diag}\left\{(\lambda_2 - \lambda_1)^{-1}, 0, (\lambda_2 - \lambda_3)^{-1}, \cdots, (\lambda_2 - \lambda_n)^{-1}\right\}.$$

From the theorem, if the perturbation matrix Δ_A can be estimated, the corresponding perturbation in the eigenvectors of A can be estimated (by first order perturbation). The steps 1), 2), and 3) in the algorithm need to find eigenvectors of the corresponding matrices. The problem now is to estimate the perturbation of the corresponding matrices from the perturbation in the image coordinates. Again we use the first order approximation to estimate these perturbations in the matrices.

For Step 1): Assume the components of the image vector $\boldsymbol{X}_i = (u_i, v_i, 1)^T$ and $\boldsymbol{X}_i' = (u_i', v_i', 1)^T$ have errors. (The third component 1 in the image vectors is accurate.) Let u_i, v_i, u_i', and v_i' have additive errors δ_{u_i}, δ_{v_i}, $\delta_{u_i'}$ and $\delta_{v_i'}$, respectively, for $1 \le i \le n$. From (2.8) we get:

$$\Delta_A^T = \begin{bmatrix} \delta_{u_1}u_1' + \delta_{u_1'}u_1 & \delta_{u_2}u_2' + \delta_{u_2'}u_2 & \cdots & \delta_{u_n}u_n' + \delta_{u_n'}u_n \\ \delta_{u_1}v_1' + \delta_{v_1'}u_1 & \delta_{u_2}v_2' + \delta_{v_2'}u_2 & \cdots & \delta_{u_n}v_n' + \delta_{v_n'}u_n \\ \delta_{u_1} & \delta_{u_2} & \cdots & \delta_{u_n} \\ \delta_{v_1}u_1' + \delta_{u_1'}v_1 & \delta_{v_2}u_2' + \delta_{u_2'}v_2 & \cdots & \delta_{v_n}u_n' + \delta_{u_n'}v_n \\ \delta_{v_1}v_1' + \delta_{v_1'}v_1 & \delta_{v_2}v_2' + \delta_{v_2'}v_2 & \cdots & \delta_{v_n}v_n' + \delta_{v_n'}v_n \\ \delta_{v_1} & \delta_{v_2} & \cdots & \delta_{v_n} \\ \delta_{u_1'} & \delta_{u_2'} & \cdots & \delta_{u_n'} \\ \delta_{v_1'} & \delta_{v_2'} & \cdots & \delta_{v_n'} \\ 0 & 0 & \cdots & 0 \end{bmatrix}. \qquad (3.20)$$

Assume the errors between the different points and different components in the image coordinates are uncorrelated, and they have the same variance σ^2 (general cases with correlation can be formulated in a similar way). With this assumption we get

$$\Gamma_{A^T} = \sigma^2 \,\text{diag}\,\{P_1, P_2, \cdots, P_n\} \qquad (3.21)$$

where P_i, $1 \le i \le n$, is a 9 by 9 submatrix:

$$P_i = \begin{bmatrix} \boldsymbol{X}_i'\boldsymbol{X}_i'^T & 0 & 0 \\ 0 & \boldsymbol{X}_i'\boldsymbol{X}_i'^T & 0 \\ 0 & 0 & \boldsymbol{X}_i'\boldsymbol{X}_i'^T \end{bmatrix} + \begin{bmatrix} u_iu_iJ & u_iv_iJ & u_iJ \\ u_iv_iJ & v_iv_iJ & v_iJ \\ u_iJ & v_iJ & J \end{bmatrix} \qquad (3.22)$$

where

$$J = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \qquad (3.23)$$

Consider the error in $\boldsymbol{h}$ in (2.10). From the Theorem and (2.9), we have (note that $\boldsymbol{h}$ is an eigenvector of A^TA instead of A):

$$\begin{aligned}\boldsymbol{\delta}_h &\cong H\Delta H^T\Delta_{A^TA}\boldsymbol{h}\\ &= H\Delta H^T[h_1I_9\; h_2I_9\; \cdots\; h_9I_9]\boldsymbol{\delta}_{A^TA}\\ &\triangleq G_h\boldsymbol{\delta}_{A^TA}. \end{aligned}\tag{3.24}$$

In the above equations, we have rewritten the matrices Δ_{A^TA} by $\boldsymbol{\delta}_{A^TA}$ and moved the perturbation to the right. In this way, the perturbation of the eigenvector is then the linear transformation (by matrix G_h) of the perturbation vector $\boldsymbol{\delta}_{A^TA}$. We have Γ_{A^T} $(= \Gamma_A^T)$ in (3.21). We need to relate $\boldsymbol{\delta}_{A^TA}$ in (3.24) to $\boldsymbol{\delta}_{A^T}$. Similar to (3.8), using first order approximation, we get

$$\Delta_{A^TA} \cong A^T\Delta_A + \Delta_A^TA. \tag{3.25}$$

Let

$$A^T = [a_{ij}]^T \triangleq [A_1\, A_2\, \cdots\, A_n] \tag{3.26}$$

we write

$$\boldsymbol{\delta}_{A^TA} \cong G_{A^TA}\boldsymbol{\delta}_{A^T} \tag{3.27}$$

where G_{A^TA} can be easily determined from (3.25):

$$G_{A^TA} = [F_{ij}] + [G_{ij}] \tag{3.28}$$

where $[F_{ij}]$ and $[G_{ij}]$ are matrices with 9 by n submatrices F_{ij} and G_{ij}, respectively. $F_{ij} = a_{ij}I_9$. G_{ij} is a 9 by 9 matrix with the ith column being the column vector $\boldsymbol{A}_j$ (see 3.26) and all other columns being zeros. From (3.24) and (3.27) we get

$$\boldsymbol{\delta}_h \cong G_h\boldsymbol{\delta}_{A^TA} \cong G_hG_{A^TA}\boldsymbol{\delta}_{A^T} \triangleq D_h\boldsymbol{\delta}_{A^T}. \tag{3.29}$$

Then

$$\Gamma_h \cong D_h\Gamma_{A^T}D_h^T. \tag{3.30}$$

From (2.11) we get the covariance matrix for E:

$$\Gamma_E = 2\Gamma_h \cong 2D_h\Gamma_{A^T}D_h^T. \tag{3.31}$$

Starting from the covariance matrix of the perturbation in A^T, we get the covariance matrix of the perturbation in the eigenvector of A^TA. This is done for E in (2.11). For $\boldsymbol{T}_s$ in (2.12) and for $\boldsymbol{q}$ in (2.19) and (2.35), the approaches are similar. For the perturbation vectors of the remaining parameters, we get the linear expression in terms of $\boldsymbol{\delta}_E$. For example, if we get D_{T_s} such that $\boldsymbol{\delta}_{T_s} \cong D_{T_s}\boldsymbol{\delta}_E$, we have $\Gamma_{T_s} \cong D_{T_s}\Gamma_ED_{T_s}^T$.

The solution of step 1) needs the eigenvector of A^TA associated with the smallest eigenvalue. The smallest eigenvalue is a simple zero eigenvalue when rank $\{A\} = 8$ (nondegenerate configuration), When rank $\{A\} < 8$ (i.e., when degenerate configurations occur), the solution $\boldsymbol{h}$ in step 1) is very sensitive to noise. As can be seen from (3.18), the second diagonal entry of Δ is infinite when $\lambda_1 = \lambda_2$. This makes the estimated errors infinite.

However, in most real applications, we do not know the noise-free A. We only know the noise-corrupted A: $A(\epsilon)$. We have to use $A(\epsilon)$ to estimate A. In the presence of noise, generally, the rank of $A(\epsilon)$ is full mathematically and the smallest eigenvalue of $A(\epsilon)^TA(\epsilon)$ is a small positive value. If noise is reasonably small, when rank $(A) < 8$ we have $\lambda_1 \approx \lambda_2$. Then large estimates of errors are still generated. From a slightly different point of view, we can regard A as a "noise-corrupted" matrix by adding $-\Delta_A$ to the matrix $A(\epsilon)$. Now the error is the deviation of the true solution from the noise-corrupted solution. This observation justify our use of the noise-corrupted A to estimate errors.

For Step 2): $\boldsymbol{T}_s$ is the unit eigenvector of EE^T associated with the smallest eigenvalue. As we did earlier we need $\boldsymbol{\delta}_{EE^T}$ to use the theorem. From

$$\Delta_{EE^T} \cong E\Delta_E^T + \Delta_EE^T \tag{3.32}$$

it is easy to find D_{EE^T} such that

$$\boldsymbol{\delta}_{EE^T} \cong D_{EE^T}\boldsymbol{\delta}_E. \tag{3.33}$$

In fact,

$$D_{EE^T} = [F_{ij}] + [G_{ij}] \tag{3.34}$$

where $[F_{ij}]$ and $[G_{ij}]$ are matrices with 3 by 3 submatrices F_{ij} and G_{ij}, respectively. F_{ij} is a 3 by 3 matrix with the ith column being $\boldsymbol{E}_j$ [see (2.11)] and all other columns being zeros. $G_{ij} = e_{ij}I_3$.

Using the theorem, we have (we use the same letter Δ and H to avoid introducing new letters)

$$\begin{aligned}\boldsymbol{\delta}_{T_s} &\cong H\Delta H^T\Delta_{EE^T}\boldsymbol{T}_s \cong H\Delta H^T[s_1I_3\; s_2I_3\; s_3I_3]\boldsymbol{\delta}_{EE^T}\\ &= H\Delta H^T[s_1I_3\; s_2I_3\; s_3I_3]D_{EE^T}\boldsymbol{\delta}_E \triangleq D_{T_s}\boldsymbol{\delta}_E \end{aligned}\tag{3.35}$$

where $\boldsymbol{T}_s = (s_1, s_2, s_3)^T$. $\boldsymbol{\delta}_{\hat{T}}$ is the same as $\boldsymbol{\delta}_{T_s}$ except with a proper sign change depending on (2.13). we get

$$\Gamma_{T_s} = \Gamma_{\hat{T}} = D_{T_s}\Gamma_ED_{T_s}^T. \tag{3.36}$$

For conciseness, we define a new vector $\boldsymbol{K}$ that combines the vector $\boldsymbol{T}_s$ with the vector $\boldsymbol{E}$:

$$\boldsymbol{K} \triangleq \begin{bmatrix}\boldsymbol{T}_s\\ \boldsymbol{E}\end{bmatrix} = \begin{bmatrix}\boldsymbol{T}_s\\ \boldsymbol{E}_1\\ \boldsymbol{E}_2\\ \boldsymbol{E}_3\end{bmatrix}. \tag{3.37}$$

So,

$$\boldsymbol{\delta}_K \cong \begin{bmatrix}D_{T_s}\\ I_9\end{bmatrix}\boldsymbol{\delta}_E \triangleq D_K\boldsymbol{\delta}_E. \tag{3.38}$$

For Step 3): From (2.20) and using first order approximation, we have

$$\begin{aligned}B(\epsilon) = B + \Delta_B &= \sum_{i=1}^{3}(B_i + \Delta_{B_i})^T(B_i + \Delta_{B_i})\\ &\cong \sum_{i=1}^{3}B_i^TB_i + \sum_{i=1}^{3}(B_i^T\Delta_{B_i} + \Delta_{B_i}^TB_i) \end{aligned}\tag{3.39}$$

where

$$\Delta_B \cong \sum_{i=1}^{3} (B_i^T \Delta_{B_i} + \Delta_{B_i}^T B_i). \tag{3.40}$$

If we use (2.18) to solve for R, using (2.17) we have

$$\begin{aligned}
\boldsymbol{\delta}_{W_1} &\cong \boldsymbol{E}_1 \times \boldsymbol{\delta}_{T_s} - \boldsymbol{T}_s \times \boldsymbol{\delta}_{E_1} - \boldsymbol{E}_3 \times \boldsymbol{\delta}_{E_2} + \boldsymbol{E}_2 \times \boldsymbol{\delta}_{E_3}\\
\boldsymbol{\delta}_{W_2} &\cong \boldsymbol{E}_2 \times \boldsymbol{\delta}_{T_s} - \boldsymbol{T}_s \times \boldsymbol{\delta}_{E_2} - \boldsymbol{E}_1 \times \boldsymbol{\delta}_{E_3} + \boldsymbol{E}_3 \times \boldsymbol{\delta}_{E_1}\\
\boldsymbol{\delta}_{W_3} &\cong \boldsymbol{E}_3 \times \boldsymbol{\delta}_{T_s} - \boldsymbol{T}_s \times \boldsymbol{\delta}_{E_3} - \boldsymbol{E}_2 \times \boldsymbol{\delta}_{E_1} + \boldsymbol{E}_1 \times \boldsymbol{\delta}_{E_2}
\end{aligned} \tag{3.41}$$

or

$$\begin{aligned}
\boldsymbol{\delta}_W &\cong \begin{bmatrix} [\boldsymbol{E}_1]_\times & -[\boldsymbol{T}_s]_\times & -[\boldsymbol{E}_3]_\times & [\boldsymbol{E}_2]_\times \\ [\boldsymbol{E}_2]_\times & [\boldsymbol{E}_3]_\times & -[\boldsymbol{T}_s]_\times & -[\boldsymbol{E}_1]_\times \\ [\boldsymbol{E}_3]_\times & -[\boldsymbol{E}_2]_\times & [\boldsymbol{E}_1]_\times & -[\boldsymbol{T}_s]_\times \end{bmatrix} \boldsymbol{\delta}_K \\
&\triangleq G_W \boldsymbol{\delta}_K \cong G_W D_K \boldsymbol{\delta}_E \triangleq D_W \boldsymbol{\delta}_E.
\end{aligned} \tag{3.42}$$

Letting $W = [w_{ij}]$, from (3.40) we get

$$\boldsymbol{\delta}_B \cong G_B \boldsymbol{\delta}_W \cong G_B D_W \boldsymbol{\delta}_E \triangleq D_B \boldsymbol{\delta}_E \tag{3.43}$$

where

$$G_B = 2 \begin{bmatrix} F_t \\ F_b \end{bmatrix} \tag{3.44}$$

where

For the case when (2.16) is used to solve for R, the discussion of the relationship between $\boldsymbol{\delta}_B$ and $\boldsymbol{\delta}_E$ [corresponding to (3.43)] is relegated to Appendix C.

Having now the expression of $\boldsymbol{\delta}_B$, we are ready to give the covariance matrix of $\boldsymbol{q}$. Since $\boldsymbol{q}$ is a unit eigenvector of B associated with the smallest eigenvalue, using the theorem, we have

$$\begin{aligned}
\boldsymbol{\delta}_q &\cong H\Delta H^T \Delta_B \boldsymbol{q} = H\Delta H^T [q_0 I_4 \; q_1 I_4 \; q_2 I_4 \; q_3 I_4] \boldsymbol{\delta}_B \\
&\cong H\Delta H^T [q_0 I_4 \; q_1 I_4 \; q_2 I_4 \; q_3 I_4] D_B \boldsymbol{\delta}_E \triangleq D_q \boldsymbol{\delta}_E.
\end{aligned} \tag{3.45}$$

Using the relation between $\boldsymbol{q}$ and R, and (2.22), we get the first order perturbation vector of R:

$$\begin{aligned}
\boldsymbol{\delta}_R &\cong 2 \begin{bmatrix} q_0 & q_1 & -q_2 & -q_3 \\ q_3 & q_2 & q_1 & q_0 \\ -q_2 & q_3 & -q_0 & q_1 \\ -q_3 & q_2 & q_1 & -q_0 \\ q_0 & -q_1 & q_2 & -q_3 \\ q_1 & q_0 & q_3 & q_2 \\ q_2 & q_3 & q_0 & q_1 \\ -q_1 & -q_0 & q_3 & q_2 \\ q_0 & -q_1 & -q_2 & q_3 \end{bmatrix} \boldsymbol{\delta}_q \\
&\triangleq G_R \boldsymbol{\delta q} \cong D_R D_q \boldsymbol{\delta}_E \triangleq D_R \boldsymbol{\delta}_E.
\end{aligned} \tag{3.46}$$

$$F_t = \begin{bmatrix}
w_{11} - 1 & w_{21} & w_{31} & w_{12} & w_{22} - 1 & w_{32} & w_{13} & w_{23} & w_{33-1} \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 & 0 \\
w_{11} - 1 & w_{21} & w_{31} & w_{12} & w_{22} + 1 & w_{32} & w_{13} & w_{23} & w_{33+1} \\
0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0
\end{bmatrix}$$

$$F_b = \begin{bmatrix}
0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
w_{11} + 1 & w_{21} & w_{31} & w_{12} & w_{22} - 1 & w_{32} & w_{13} & w_{23} & w_{33+1} \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\
w_{11} + 1 & w_{21} & w_{31} & w_{12} & w_{22} + 1 & w_{32} & w_{13} & w_{23} & w_{33-1}
\end{bmatrix}$$

As in step 1), in steps 2) and 3) we estimate the errors by using the perturbed E and B to substitute the noise-free E and B.

In summary, the perturbation vectors of the parameters $\boldsymbol{q}$ and R are expressed in terms of linear transformation of perturbation of E [(3.45), (3.46)]. The covariance matrix of the perturbation of E is given in (3.31). The covariance matrix of $\boldsymbol{q}$ and R are then

$$\Gamma_q = D_q \Gamma_E D_q^T \tag{3.47}$$

$$\Gamma_R = D_R \Gamma_E D_R^T. \tag{3.48}$$

From the covariance matrix of the perturbation, we can estimate the Euclidean norm of the perturbation vector and the perturbation matrix by the square root of the trace of the corresponding covariance matrix of the perturbations.

$$\| \delta_{\hat{T}} \| \approx \sqrt{\text{trace } \{ \Gamma_{\hat{T}} \}} \tag{3.49}$$

$$\| \Delta_R \| = \| \delta_R \| \approx \sqrt{\text{trace } \{ \Gamma_R \}} \tag{3.50}$$

Similarly we get estimate of perturbation in $\boldsymbol{q}$. Since the Euclidean norm of the orthonormal matrix R is equal to $\sqrt{3}$, the relative perturbation in R is defined by $\| \Delta_R \| / \sqrt{3}$.

The problem of estimating errors in the relative depths can be formulated in a similar manner. However, as shown by the simulation, the variances of the errors in the relative depths are considerably larger than those of the motion parameters. This is reasonable, since for each 3-D point, we just get two observations. Therefore the estimated mean of the errors in depths based on two images is not a good estimate of the actual errors.

IV. Errors Versus Structure, Motion, and System Parameters

In reality the perspective projections of feature points are corrupted by noise. The noise includes the feature detector errors, matching errors, quantization errors and system calibration errors. All those errors result in errors in the solution of the motion parameters and 3-D structure of the scene. It is observed that computer roundoff errors are generally far less significant than those mentioned above, provided a double precision (about 64 bits) is used for real numbers. So we assume the noise is introduced solely through the perturbations in the image coordinates of the projection of feature points.

However, with the same noise level, the resulting errors are not always the same for different scene structure, different motion, and different system setups. The question is how they are related and to what degree they affect the reliability of the estimates. The factors we will discuss that affect the reliability of the estimates fall into three categories:

1) structure of the scene,

2) motion,

3) parameters of imaging systems.

Our analysis is mainly based on the algorithm presented in this paper. We also provide algorithm-independent perspectives.

A. Structure of the Scene

The 9-dimensional unit vector $\boldsymbol{h}$ is determined up to a sign if and only if the rank of A in (2.10) is equal to 8. A necessary and sufficient condition for the rank of A to equal 8 is given by Longuet-Higgins [21]. Assuming the relative motion is due to motion of the camera, the condition is that the feature points do not lie on any quadratic surface that passes through the projection center of the camera at the two time instants. To satisfy this condition, at least 8 points are required. More points are needed for yielding overdetermination to combat noise. If a set of feature points is such that the rank of corresponding A is less than 8, we say that the structure is degenerate. To ensure that the structure is far from degenerate, intuitively, the structure of the points should be such that they are very irregular.

In the presence of noise, the rank of A is generally mathematically full even if the actual structure is degenerate. If the structure is nearly degenerate, the solution of (2.10) is conceivably not reliable. So, in the presence of noise, we should consider the numerical condition of the matrix A. The previous section presents a method to determine such a condition and gives an estimate of the errors.

Obviously, if the cluster of projections of feature points is confined in a small portion of images, only a small portion of the image resolution is used. This will certainly result in less reliable solutions. So, the configuration of the feature points should be such that its projection covers as much of the images as possible.

In the discussion of Section IV-B we will see that long displacement vectors will result in more reliable solutions. For the same amount of motion, the scene should be close to the camera so that it yields long displacement in image plane. This condition is actually related to the numerical condition of matrix A.

Another factor is the number of feature points. It is very effective to reduce the error in the solutions by using more points in addition to the minimally required 8. Since a severely noise-corrupted image vector can pull the solution away from the correct one by a large amount, it is desirable to use only reliable matches (information given by, e.g., a point matcher) for motion parameter estimation.

It is clear that the relative depths can be reliably determined by (2.25) only if $\boldsymbol{X}_i'$ and $R\boldsymbol{X}_i$ are linearly independent. That is, $\boldsymbol{X}_i' \times R\boldsymbol{X}_i \neq 0$. When $\boldsymbol{T} \neq 0$, all points satisfies this except at most one. In fact $\boldsymbol{X}_i' \times R\boldsymbol{X}_i = 0$ if and only if $\boldsymbol{T} \times R\boldsymbol{X}_i = 0$ from (2.5). Let $\boldsymbol{X}_p$ be such that $\boldsymbol{T} \times R\boldsymbol{X}_p = 0$. If $\boldsymbol{X}_p$ happens to be the image vector of a feature point, the depth of this point can not be determined. For those points whose projections are close to $\boldsymbol{X}_p$, the corresponding depths can not be reliably determined in the presence of noise. If rotation angle is equal to zero, the point $\boldsymbol{X}_p$ corresponds to the focus of expansion or contraction. At this point, the projection is the same before and after motion.

B. Motion Parameters

As mentioned earlier, a motion can be represented by a rotation followed by a translation.

Magnitude of Translation: If the magnitude of a translation vector is equal to zero, the solution of the translation direction is arbitrary (since $\boldsymbol{T}_s$ in (2.26) is an arbitrary unit vector) and the depths of the feature points cannot be determined. When $\| \boldsymbol{T} \|$ is close to zero, the direction of translation, $\hat{\boldsymbol{T}}$, cannot be reliably determined and therefore, neither can the depths of feature points [notice $\hat{\boldsymbol{T}}$ in (2.25)].

When $\boldsymbol{T} = 0$, the rank of A in (2.8) is always no larger than 6 ($E = [\boldsymbol{T}_s]_\times R$ has two degrees of freedom). R can still be determined by picking up any $\boldsymbol{h}$ satisfying (2.10), since in the definition of E in $E = [\boldsymbol{T}_s]_\times R$, $\boldsymbol{T}_s$ is just a unit vector satisfying (2.26).

Direction of Translation: This is the most interesting factor associated with the reliability of the solutions. From (2.28) and the algorithm, it can be seen that the translation direction is determined through the fact that $\boldsymbol{T} \cdot (\boldsymbol{X}' \times R\boldsymbol{X}) = 0$, or in other words, $\boldsymbol{T}$ is orthogonal to the cross product $\boldsymbol{X}' \times R\boldsymbol{X}$. Fig. 2(a) illustrates the spatial relations between three vectors $\boldsymbol{x}'$, $\boldsymbol{x}$, and $\boldsymbol{T}$. Figs. 2(b) and (c) show the cases where the translation vector is orthogonal and parallel, respectively, to the image plane. Fig. 2(d) describes a general case. Generally the projections of feature points cover considerably large area of the image, around the optical axis ($\boldsymbol{Z}$-axis). In the case of Fig. 2(b) it is clear that the vectors $\boldsymbol{X}' \times R\boldsymbol{X}$ spread over the area in $\boldsymbol{X}$–$\boldsymbol{Y}$ plane around the origin (shown by a shaded area in the figure). However for the case of Fig. 2(c) the vectors $\boldsymbol{X}' \times R\boldsymbol{X}$ are confined in a small shaded area in the $\boldsymbol{X}$–$\boldsymbol{Z}$ plane. For the general case Fig. 2(d), the area of $\boldsymbol{X}' \times R\boldsymbol{X}$ is shown by the corresponding shaded area. The algorithm determines the direction of $\boldsymbol{T}$ through $\boldsymbol{T} \cdot (\boldsymbol{X}_i' \times R\boldsymbol{X}_i) = 0$ ($i = 1, 2, \cdots, n$, a subscript is added when it is necessary, otherwise it is dropped), i.e., $\boldsymbol{T}$ is orthogonal to n vectors in the shaded area. With perturbation in $\boldsymbol{X}$ and $\boldsymbol{X}'$, the product $\boldsymbol{X}' \times R\boldsymbol{X}$ will be slightly perturbed away from the original position and it may leave the plane of the shaded area. This causes the errors in the estimated $\hat{\boldsymbol{T}}$. Since the shaded area in Fig. 2(b) spreads round the origin but that in Fig. 2(c) is confined in a small area on one side of origin, statistically the former allows a more reliable estimate of $\hat{\boldsymbol{T}}$ than the latter.

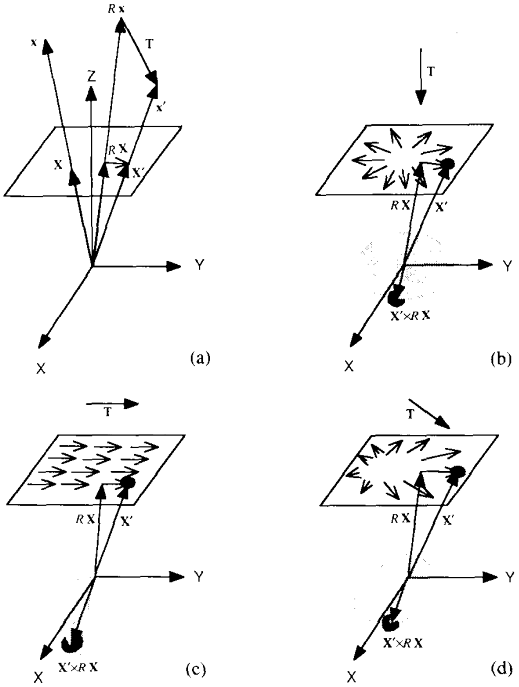


Fig. 2. Effects of perturbation versus translation direction.

On the other hand, the perturbation of Fig. 2(b) generally will not leave the shaded area as much as that of Fig. 2(c). This can be seen in the following. Assume the vector $\boldsymbol{X}'$ is perturbed in the image plane. The area of perturbation is illustrated by a small dark disk around $\boldsymbol{X}'$ in the image plane. The corresponding perturbed vector $\boldsymbol{X}' \times R\boldsymbol{X}$ is roughly represented by a small dark disk around $\boldsymbol{X}' \times R\boldsymbol{X}$. Since this disk is orthogonal to $R\boldsymbol{X}$, it is nearly parallel to the shaded area if $R\boldsymbol{X}$ is not far away from the optical axis. Similarly, the perturbation of $\boldsymbol{X}' \times R\boldsymbol{X}$ due to perturbation in $R\boldsymbol{X}$ is orthogonal to $\boldsymbol{X}'$ and so it is nearly parallel to the shaded area if $\boldsymbol{X}'$ is not far away from the optical axis. In a word, the perturbations of $\boldsymbol{X}' \times R\boldsymbol{X}$ due to individual perturbation of either $\boldsymbol{X}'$ or $R\boldsymbol{X}$ are nearly *parallel* to the shaded area if $\boldsymbol{X}'$ and $R\boldsymbol{X}$ are not far from the optical axis. Hence, such individual perturbations will not cause large errors in the estimated $\hat{\boldsymbol{T}}$. We know that perturbation of the product of two vectors is approximately the sum of the perturbations due to individual perturbations. In fact, denoting the perturbation of $\boldsymbol{a}$ by $\boldsymbol{\delta}_a$ and that of $\boldsymbol{b}$ by $\boldsymbol{\delta}_b$, we have

$$(\boldsymbol{a} + \boldsymbol{\delta}_a) \times (\boldsymbol{b} + \boldsymbol{\delta}_b) - \boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{\delta}_a \times \boldsymbol{b} + \boldsymbol{a} \times \boldsymbol{\delta}_b + \boldsymbol{\delta}_a \times \boldsymbol{\delta}_b \cong \boldsymbol{\delta}_a \times \boldsymbol{b} + \boldsymbol{a} \times \boldsymbol{\delta}_b. \qquad (4.1)$$

The last two terms in (4.1) are perturbations of $\boldsymbol{a} \times \boldsymbol{b}$ due to individual perturbation of $\boldsymbol{a}$ and $\boldsymbol{b}$, respectively. For example, $(\boldsymbol{a} + \boldsymbol{\delta}_a) \times \boldsymbol{b} - \boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{\delta}_a \times \boldsymbol{b}$.

In the case of Fig. 2(c), the corresponding perturbation of $\boldsymbol{X}' \times R\boldsymbol{X}$ due to individual perturbation of $\boldsymbol{X}'$ can be represented by a dark disk orthogonal to $R\boldsymbol{X}$ around $\boldsymbol{X}' \times R\boldsymbol{X}$ as shown in Fig. 2(c). This perturbation disk is nearly *orthogonal* to the shaded area if $R\boldsymbol{X}$ is not far from optical axis. The similar conclusion is true for the perturbation due to individual perturbation of $R\boldsymbol{X}$. Therefore statistically the perturbations of $\boldsymbol{X}'$ and $R\boldsymbol{X}$ in the case of Fig. 2(c) will cause larger errors in the estimated $\hat{\boldsymbol{T}}$ than those in the case of Fig. 2(b). It can be easily seen from Fig. 2(c) that the major perturbation of the estimated translation direction is in the $\boldsymbol{Z}$ component.

Both the shape of the shaded areas and the orientation of the perturbation disks imply that a translation orthogonal to the image plane [Fig. 2(b)] allows more stable estimation of translation direction $\hat{T}$ than a translation parallel to the image plane [Fig. 2(c)].

The relationship can be explained intuitively (algorithm independently). A translation in depth will cause less changes in the images than a translation parallel to the image plane and with the same translational magnitude. In other words, Z component of translation is very sensitive to the errors in the observed data. Therefore, Z component of the translation cannot be reliably determined. With a relative large perturbation in Z component, the direction of a translation orthogonal to image plane is not as significantly affected as the direction of a translation parallel to the image plane.

Parameters of Rotation: First, it is necessary to discuss briefly the error measurement of rotation parameters. There exist a variety of ways of representing rotation [4]. For example, 1) an axis of rotation and an angle about the axis; 2) three rotation angles about three fixed (or moving) coordinate axes, respectively; 3) rotation matrix; 4) rotation quaternion. We need a measurement for the errors of rotation that does not very much depend on the actual rotation parameters. Consider the relative error of rotation axis and rotation angle in 1). The relative error of rotation angle θ is $|\tilde{\theta} - \theta|/\theta$, where $\tilde{\theta}$ is the estimated θ. It is infinity when $\theta = 0$. If rotation angle is zero or nearly zero, the error in rotation axis is not important at all. So, the error in terms of parameters of 1) is not desirable. Similarly, relative error of the parameters in 2) is not what we need. The relative error of rotation matrix R, $\|\tilde{R} - R\|/\|R\|$ (where $\tilde{R}$ is the estimated R), and relative error of rotation quaternion do not suffer from the problems mentioned above. In this paper we use the relative error of rotation matrix to indicate the errors in the rotation parameters unless special attention is needed for rotation axis and rotation angle. Since $RI = R$, geometrically, the relative error of rotation matrix R is the square root of the mean squared distance error of the rotated orthonormal frame.

The correlation between the rotation and translation is very complicated. An exhaustive analysis is very tedious. We would rather give some perspectives.

First, we consider how rotation can be separated from translation. A rotation about the optical axis is easy to be distinguished from translation by the algorithm since no translation will give a similar displacement fields in image. How about a rotation about an axis parallel to the image plane, say the X axis? Let us consider two cases. In the first case, one rotates head about a vertical axis through his body. In the second, he translates head in the direction of rotation of the first case. If he is looking at a wall parallel to his face, the displacement field on his retina is very similar for two cases. This implies that it is difficult to tell the translation from rotation. In fact, there exist slight differences between rotation and translation in terms of projections as shown in Fig. 3. The linear algorithm employs this kind of differences since the directions of image vectors determine the essential parameter E (2.30), However, the differences are not very large, especially for short displacement vectors or at the center of images. So, the algorithm may easily confuse the translation with the rotation in the presence of noise. As a result, the solution is more sensitive to noise in the case of translation parallel to the image plane than in the case of translation orthogonal to the image plane. Similarly, rotation with a rotation axis parallel to the image plane is sensitive to noise than other rotations. However, since the displacement is mainly caused by translation in most cases, the effects caused by translation is dominant.

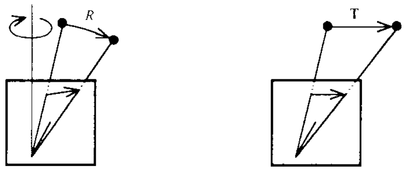


Fig. 3. Rotation and translation generate different displacement fields. Such differences are large near the peripheral areas of images.

If the translation direction cannot be reliably determined, generally rotation cannot either, since R is determined using translation. Therefore an unstable case for the estimation of translation is also unstable for the estimation of rotation.

After the translation is determined, do different rotations imply different reliabilities of the estimated rotational parameters? If we consider the relative errors in terms of rotation axis and rotation angle, different types of rotation do affect the reliability of these two parameters in a different way. As shown in Fig. 4, different perturbations in the image vectors have different amount of effects on the rotation axis $\boldsymbol{n}$ and rotation angle θ for cases (a)–(d). Figs. 4(a) and (b) correspond to the case where the rotation axis is orthogonal to the image plane. Figs. 4(c) and (d) correspond to the case where rotation axis is parallel to the image plane. The perturbation (represented by a two-way arrow) in Fig. 4(a) has smaller effect on rotation axis than that in Fig. 4(c) while both cases (a) and (c) have little effect on the rotation angle. The perturbation of Fig. 4(b) has larger effect on rotation angle than that in Fig. 4(d), while both cases (b) and (d) have little effect on rotation axis. Summarizing from Fig. 4: comparing the two cases where the rotation axis is orthogonal to the image plane and where the rotation axis is parallel to the image plane, the rotation axis can be more reliably estimated for the former and the rotation angle can be more reliably estimated for the latter.

The above opposite effects on rotation axis and rotation angle make the error in R be less sensitive to the type of motion. On the other hand, the rotation is determined after translation. The errors in the estimated translation parameters also cause errors in rotation parameters. When the errors in translation is the main reason for the errors in rotation parameters, the effects caused by different rotations are not significant. Simulations presented in Section 4 confirm that the errors in motion parameters are not sensitive to rotation parameters.

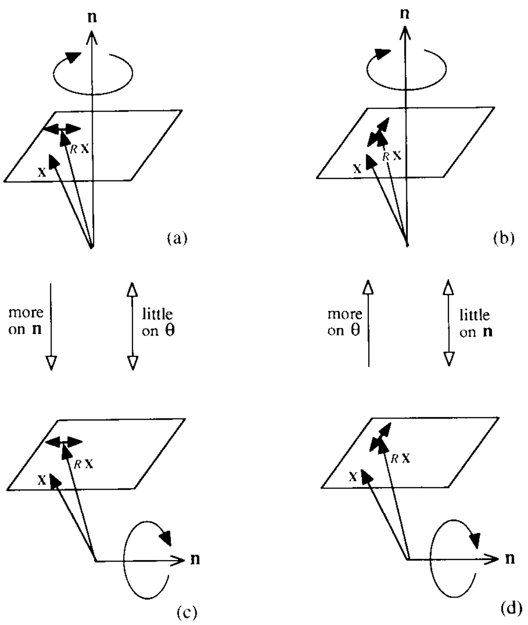


Fig. 4. Effects of perturbation versus rotation axis.

From Figs. 2, 3, and 4, it is clear that long displacement vectors will generally result in more reliable solutions than short ones in the presence of noise. To yield long displacement vectors, the motion should be large and the scene should be close to the image sensor.

C. System Parameters

Let the resolution and focal length be fixed. The remaining geometrical parameter of the imaging system of our model is the image size (equivalently field of view).

Let the image size is reduced, say by a factor of 2. For the image to cover roughly the same scene, the scene has to be moved from camera in **Z** direction such that it is about twice as far as before. This doubles the distance from the camera to the scene and reduces the variation in depth, which is an unstable factor. If the scene is not moved away, the camera will cover a smaller area of the scene, which will usually reduce the variation of depth. Furthermore, the narrowed field of view makes the portion of the scene that is visible in both images smaller (with the same amount of motion). So, equivalently, the resolution is reduced. This can be compensated by allow a smaller interframe motion. However, a small translation yields unstable estimates as we discussed earlier. In a word, a small image size (or a narrow field of view) is unstable. Another important effect of narrowing field of view is shown in Fig. 3. The field of view is crucial for distinguishing a rotation from translation, since the differences are more significant in the peripheral areas of images (see Fig. 3). So, a reduction in image size will particularly worsen the performance in the cases where translation is parallel to the image plane.

For conventional imaging sensors, the image size is fixed. The focal length is the parameter that changes the field of view. Reducing image size is equivalent to increasing focal length and vice versa.

The following section presents the statistical data through simulations to quantitatively show the relationships analyzed in this section.

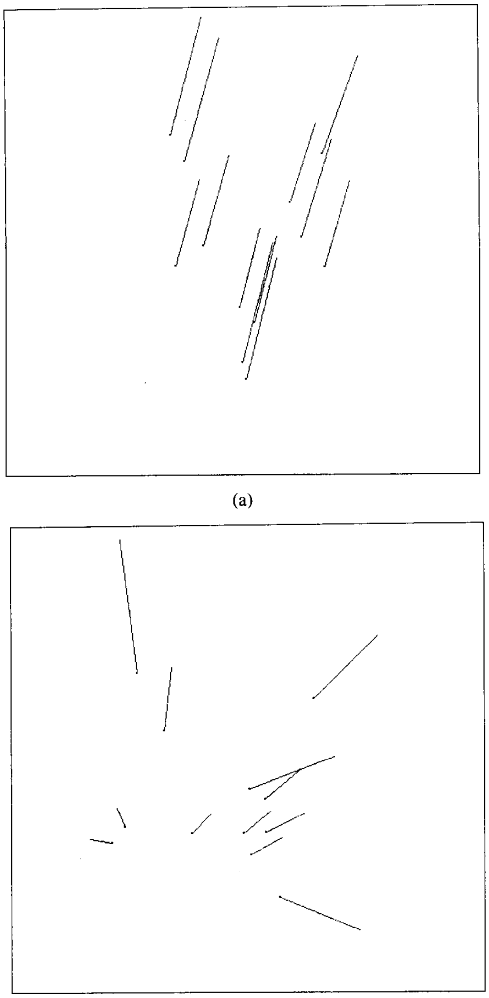
V. Experimental Results

In this section, we present the results of experiments. The first series of simulations is designed to demonstrate the performance of the motion estimation algorithm introduced in Section II, compared to the existing linear methods. The second series of simulations is to test the error estimation discussed in Section III. The third series is to show the dependency of the errors on motion and system parameters. Finally, some preliminary results are presented for images of real world scenes.

In the simulations the feature points of the scene are generated randomly according to a uniform distribution in a cube of $s \times s \times s$ (s is called *object size*) the center of which is the center of the object before motion. The distance between the projection center of the camera and center of the object is called *object distance*. The image is a square whose side length is called *image size*. The field of view is determined by the size of the image and the focal length. Those points undergo a rotation about an axis through the origin (projection center) and then a translation. Only visible points are used for the algorithm. The image coordinates of the points are quantized according to the resolution of the camera. If the resolution is m by m, the horizontal and vertical coordinates each have uniformly spaced m levels. The positions of these levels correspond to the locations of the pixels. The image coordinates are rounded off to the nearest levels before they are used by the motion estimation algorithm. These roundoff errors result in the errors in the motion parameters and the relative depths computed by the algorithm. Simulations showed that reducing the resolution by a factor of 2 roughly doubles the errors. Other additional random errors can be simulated by a reduced image resolution with the similar variance of quantization noise. All the errors shown in this section are relative errors. Relative error of a matrix, or a vector, is defined by the norm of the error matrix, or vector, divided by the Euclidean norm of the correct matrix, or vector, respectively. Since no ambiguity can arise, in the remainder of this paper, relative errors are often simply referred to as errors. Unless stated otherwise, object size is 10, object distance is 11 units, image size is 2, image resolution is 256 by 256, and the focal length is 1 unit.

A. Simulations for the Performance Improvement of the Algorithm

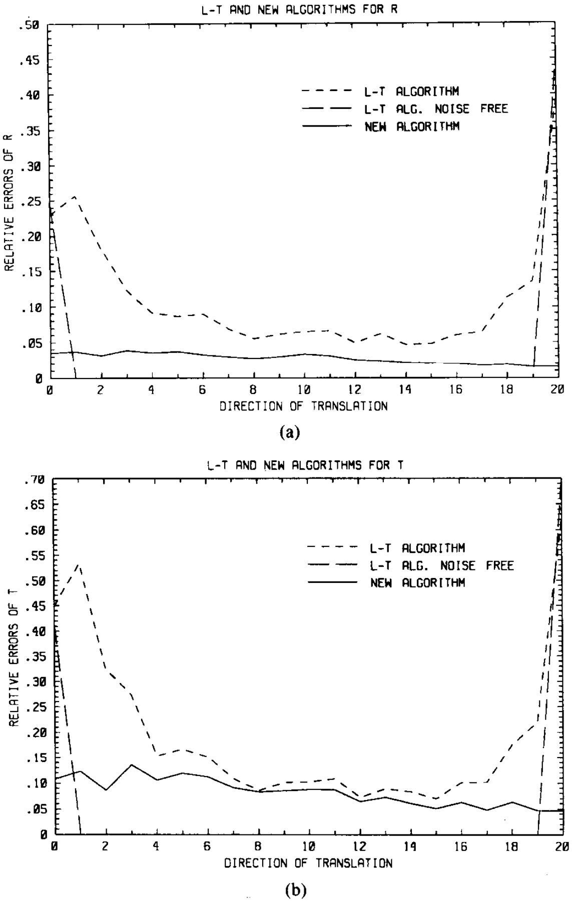
First, the errors of the algorithm are compared to an algorithm that represents typical existing linear algorithms. The algorithm by Longuet-Higgins [20] and that by Tsai and Huang [28] are two typical existing linear algorithms. Since the way to compute the ratios of the



(a)

(b)

Fig. 5. Two examples of displacement fields for the data shown in Fig. 6. (a) Horizontal index: 0. (b) Horizontal index: 20.



(a)

(b)

Fig. 6. Relative errors of L-T algorithm and our new algorithm. (a) Relative errors of *R*. (b) Relative errors of *T*. Rotation axis: $(-0.2, 1, 0.2)$. Rotation angle: 8°. For horizontal axis from 0 to 20, the projection of translation onto the $X - Z$ plane changes from the X to the $-Z$ direction (with magnitude 3.0, evenly spaced 21 translation directions). The Y component is always equal to 1.

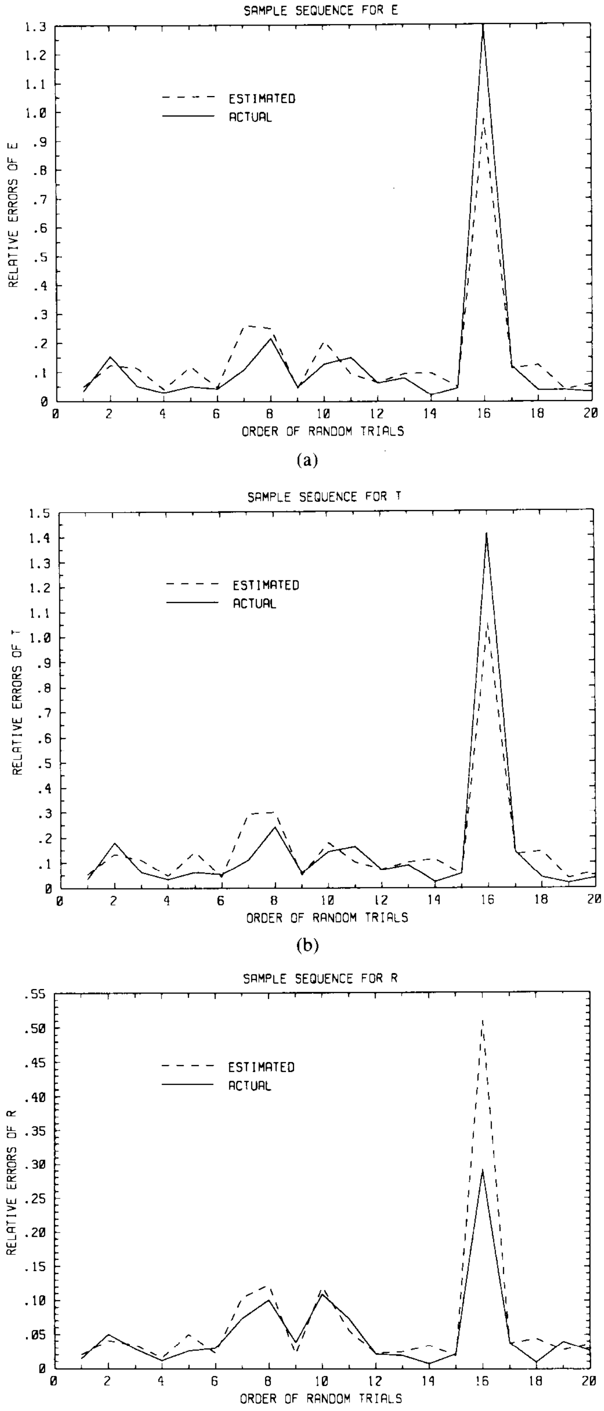
components of $\boldsymbol{T}$ from $\hat{Q}Q$ [20, equation (17)] is not specifically given, the unit vector $\hat{\boldsymbol{T}}$ is determined by the algorithm of Tsai and Hunag [28]. Rotation matrix R is determined by the method in [20], since it is computationally simpler than that in [28]. Such a composite algorithm represents the typical algorithms that are designed primarily for noise free data. We call it the L-T algorithm (Longuet-Higgins and Tsai). We compare the performance of our new algorithm to the L-T algorithm on arbitrarily chosen motion parameters. In Fig. 6 the rotation is about an axis $(-0.2, 1, 0.2)$ by an angle 8°. The projection of translation onto the $X - Z$ plane changes from the X to the $-Z$ direction (with magnitude 3.0, evenly spaced 21 translation directions from the X to the $-Z$ in $X - Z$ plane, with horizontal index from 0 to 20). The Y component is always equal to 1. Twelve point correspondences are used. Fig. 5 shows two examples of displacement fields, corresponding to horizontal indexes 0 and 20, respectively. Fig. 6(a) shows the errors of R and Fig. 6(b) shows that of $\boldsymbol{T}$, averaging through 100 trials (random trial in the following always means randomly generated points). Significant improvement of our new algorithm over the L-T algorithm can be seen. Fig. 6 also shows the errors of the L-T algorithm for the same motion parameters, but no roundoff is performed for image coordinates (noise-free). Idealy the error should be almost equal to zero. However, this is not so for horizontal indexes 0 and 20. The reason for this can be easily seen from [28, equations (22)-(27)]. In other words, some special cases are not considered in [28]. This partially accounts for the different amount of improvement for different translations.

The performance of the algorithm in other cases is demonstrated in the remainder of this section.

B. Simulations for Error Estimation

For error estimation, we assume the roundoff errors are uniformly distributed between plus half and minus half of the pixel size. So the variance of the errors in the image coordinates is $\sigma^2 = p^2/12$, where p is the spacing between the quantization levels in the images (pixel size). This variance is used for error estimation. Different motion parameters with different image resolutions were simulated. The errors in solution for motion parameters were reduced roughly by a factor of two when the image resolution was doubled. Fig. 7 shows the results of a typical sequence of trials with 9 point correspondences. Twenty random trials are shown in the order of their generation. Fig. 7(a) shows error estimation for relative errors of E. As can be seen from the figure, the estimated errors are strongly correlated with the actual errors. The estimated errors are especially important to detect a nearly degenerate configuration (e.g., trial No. 16 in Fig. 7(a) where relatively unreliable results are generated by the algorithm). Figs. 7(b) and (c) show the relative errors in the translation $\boldsymbol{T}$ and the rotation matrix $\boldsymbol{R}$, respectively. The very similar curves of errors in E, $\boldsymbol{T}$, and $\boldsymbol{R}$ indicate that the steps after 1) are stable. The main errors are from the estimates of E. In other words, the accuracy of E dominates the accuracy of the final motion parameters.

The average performances of the error estimation and our motion estimation algorithm are presented in Fig. 8. Average relative errors (solid curves) are recorded through 20 random trials, with the same motion as that in Fig. 7, for different numbers of point correspondences used. (the sequence with 9 point correspondences is presented in Fig. 7). In Fig. 8 the long-dashed curves indicate the mean absolute difference between the estimated error and the actual error (called deviation of error estimation here), and the short-dashed curves indicate the bias (difference between the mean of the estimated errors and the mean of the actual errors) through these 20 trials. As can been seen from Fig. 8, the errors decrease very quickly when the number of points increases beyond the required minimum of 8. This indicates that it is very effective to reduce the error by using a few more points in addition to the minimally required 8. It can also be seen that the mean deviation between the estimated error and the actual error is about a half of the actual error with the exception of the cases where the number of points is equal to 8. When the number of point correspondences is 8, there is a reasonably high probability for the randomly generated points to form a nearly degenerate configuration. When the point configuration is degenerate or nearly degenerate, the difference between the estimated error and the actual error is large. This is one of the reasons for the large deviations and bias in the 8-point case. Some individual simulations still show a good agreement between the estimated errors and the actual errors in the 8-point case. Fig. 9 shows the mean relative errors in the relative depths with the same motion averaging through 100 trials.



(a)

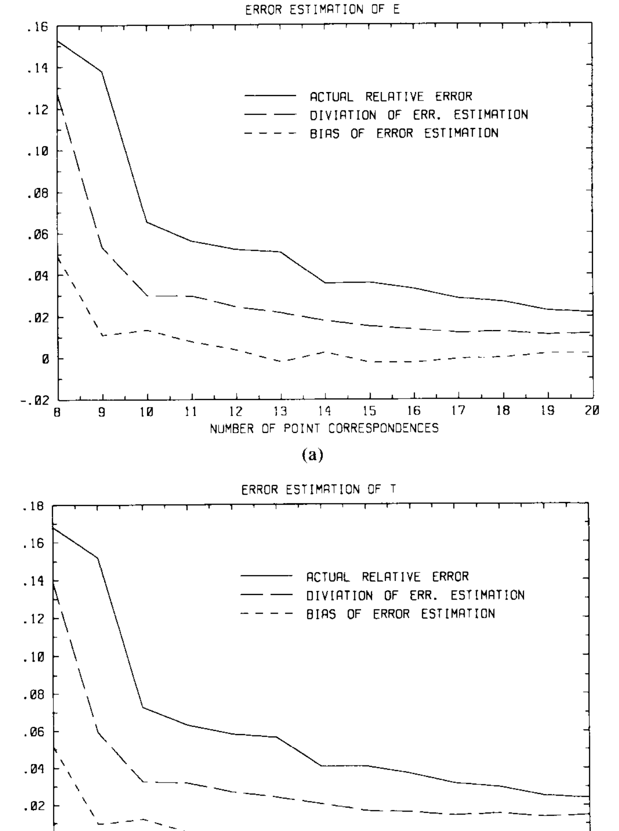
(b)

(c)

Fig. 7. Actual relative errors and estimated relative errors of (a) E, (b) $\boldsymbol{T}$, and (c) R. Rotation axis: (1, 0.9, 0.8). Rotation angle: 5°. Translation: (0.5, −0.5, −3.0).

C. Simulations for Error Depending on Motion and System Parameters

The experiments show significant dependencies of the errors in the motion parameters on the values of the motion and system parameters. The stability of the structure can be estimated using the approach introduced in Section III. So, the effect of structure is excluded here. For the simulations presented in this subsection, 12 feature points are used in each image (i.e., 12 point correspondences). One hundred trials are recorded for the average relative errors.



(a)

(b)

(c)

Fig. 8. Statistical record of error estimation. Actual relative error, deviation of error estimation, and bias of error estimation for (a) E, (b) T, and (c) R versus number of point correspondences. Rotation axis: (1, 0.9, 0.8). Rotation angle: 5°. Translation: (0.5, −0.5, −3.0).

1) *Motion Parameters:*

Magnitude of Translation: With rotation axis (1, 0, 0) and rotation angle 5°, the translation direction being equal to $(k, 0, k)$ where k is such that the length of translation vector changes from 0.5 up to 4.5 with 20 evenly spaced values, the average relative errors of the estimates are shown in Fig. 10. It is very clear that the errors in rotation matrix R is almost unaffected by merely changing the magnitude of translation. However the errors in translation direction and relative depths drastically decrease as the magnitude of translation increases. This is consistent with the discussion in Section IV.

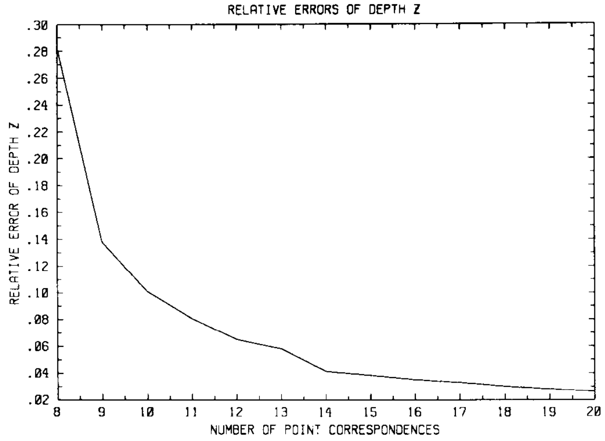


Fig. 9. Relative error of relative depths versus number of point correspondences. Same motion as in Fig. 8.

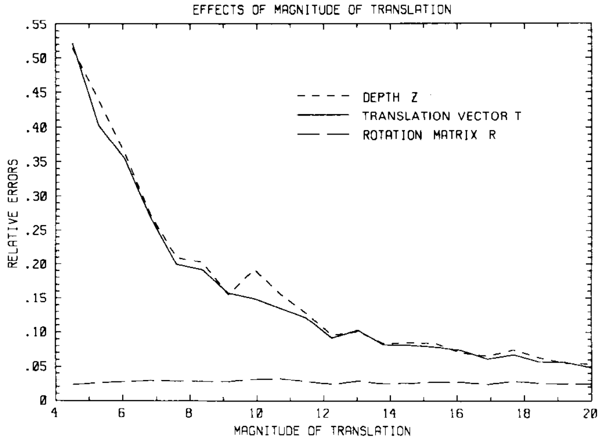
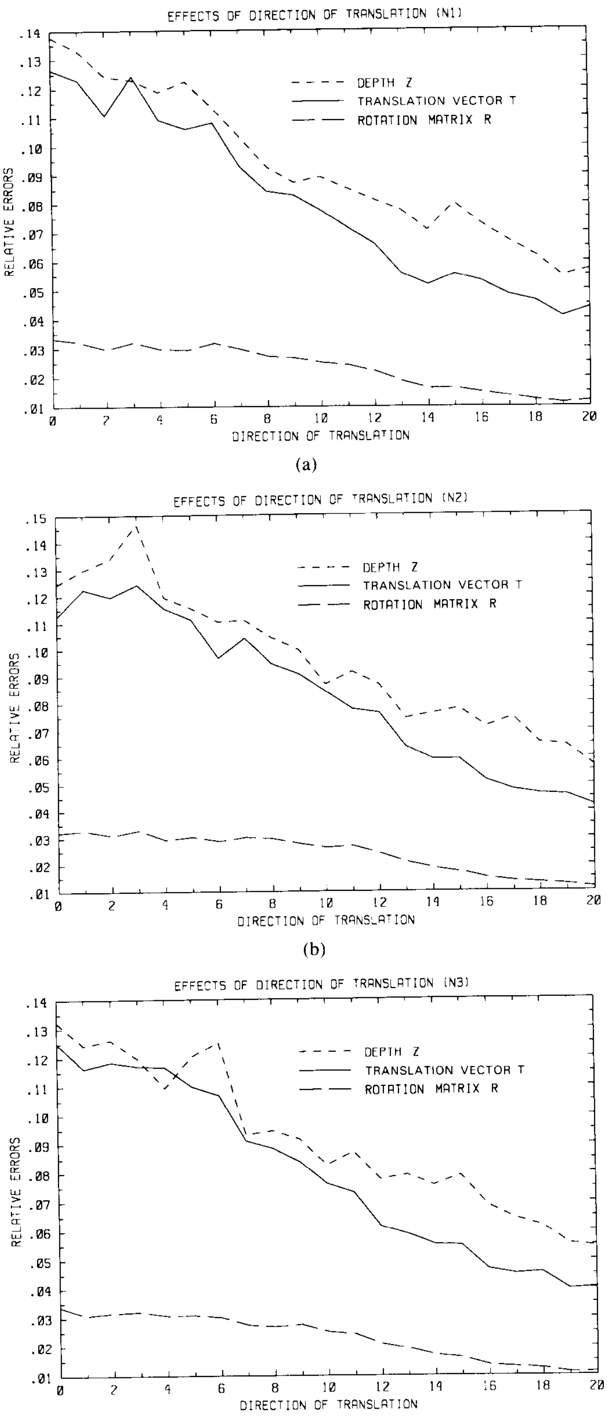


Fig. 10. Relative errors versus magnitude of translation. Rotation axis: (1, 0, 0). Rotation angle: 5°. Translation direction direction: $(k, 0, k)$. k is such that the length of the translation vector changes from 0.5 up to 4.5 with 20 evenly spaced values along the horizontal axis.

Direction of Translation: In Fig. 11, with the magnitude of translation fixed to be 3, the direction of the translation changes from (1, 0, 0) to (0, 0, 1) with evenly spaced 20 directions. The rotation angle is 5°. Three rotation axes (1, 0, 0), (0, 1, 0) and (0, 0, 1) are used in Figs. 11(a), (b), and (c), respectively. Despite of different rotation axes (it has been discussed in Section IV and will be shown soon that rotation parameters has no significant effects), the errors for rotation matrix, translation direction and relative depths all significantly decrease as the translation direction changes from being parallel to being orthogonal to image plane. The reasons for this relationship have been discussed earlier.

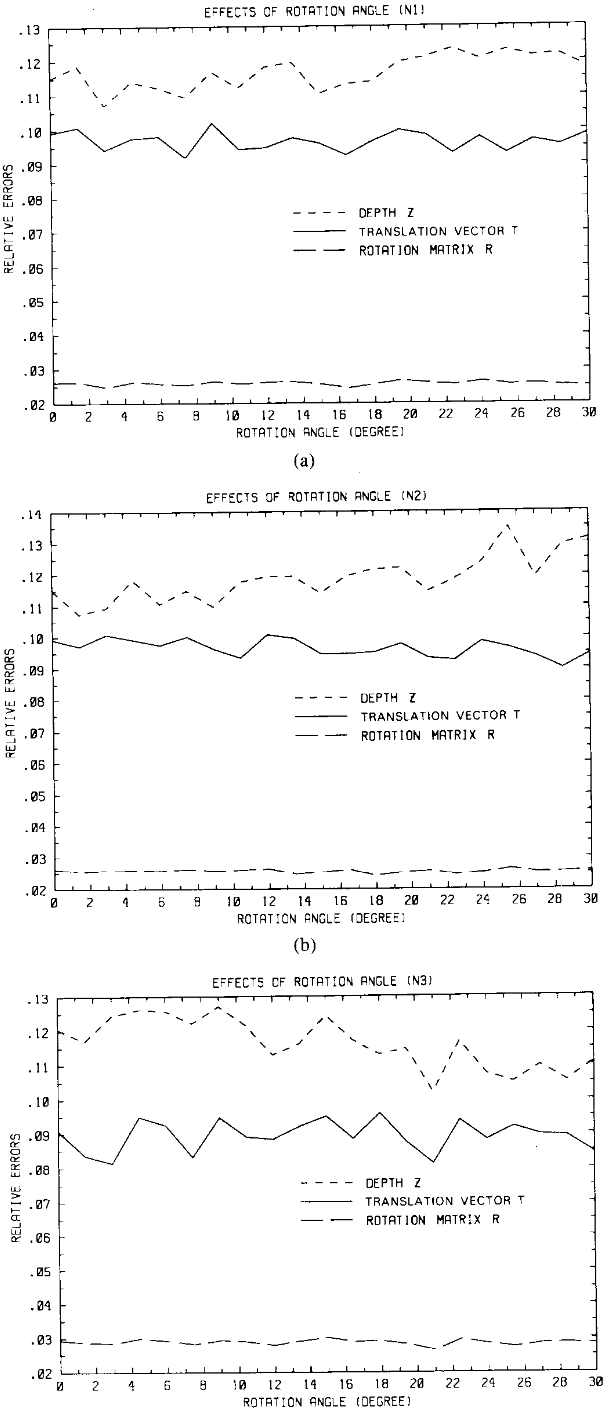


(a)

(b)

(c)

Fig. 11. Relative errors versus direction of translation. Rotation axis: (a) (1, 0, 0); (b) (0, 1, 0); (c) (0, 0, 1). Rotation angle: 5°. With horizontal index from 0 to 20, the direction of the translation changes from (1, 0, 0) to (0, 0, 1) with evenly spaced 20 directions. The magnitude of translation is fixed to be 3.



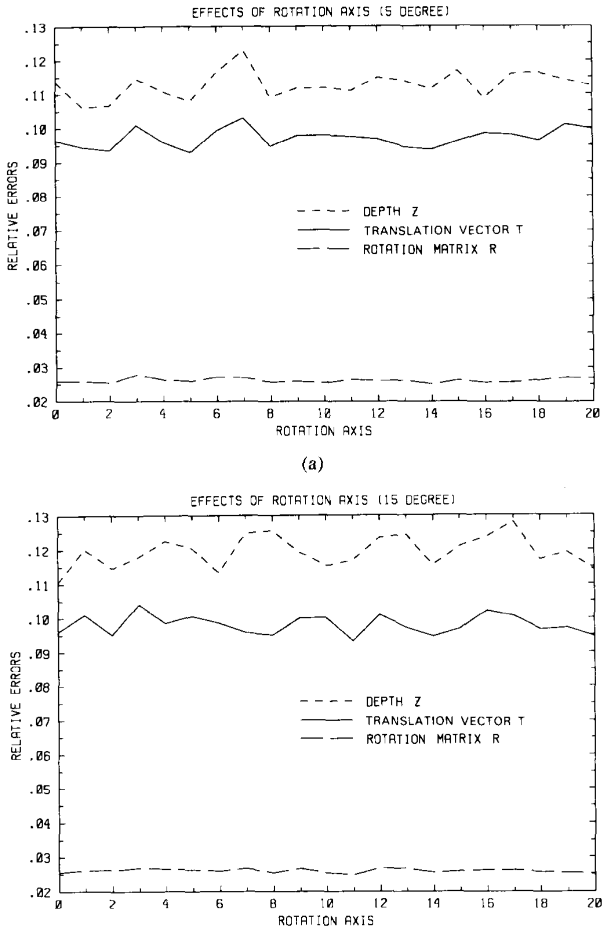
(a)

(b)

(c)

Fig. 12. Relative errors versus rotation angle. (a) Rotation axis: (1, 0, 0). Translation: (1.732, 0, −1.732). (b) Rotation axis: (0, 0, 1). Translation: (1.732, 0, −1.732). (c) Rotation axis: (1, 1, 1). Translation: (1.732, 1.732, −1.732).

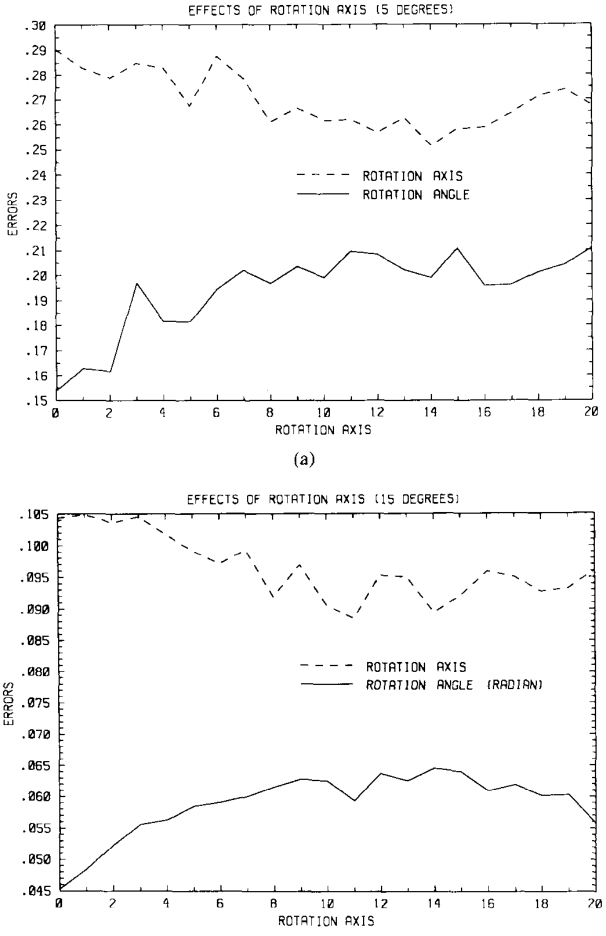
Rotation Parameters: We mentioned earlier that the rotation parameters generally do not significantly affect the errors in solutions. In Fig. 12, rotation angle changes from 0° to 30° and with different translation vectors and rotation axes (see captions of the figures). It can be seen from the figures that errors in R, $\hat{T}$ and relative depths are not significantly affected by the change of rotation angle. Similarly, Fig. 13 shows that rotation axis does not sig-



(a)

(b)

Fig. 13. Relative errors versus rotation axis. Translation: (1.732, 0, −1.732). With horizontal index from 0 to 20, rotation axis changes from (1, 0, 0) to (0, 0, 1) in the *X*-*Z* plane at 20 evenly spaced directions. Rotation angle: (a) 5°; (b) 15°.



(a)

(b)

Fig. 14. Relative errors of rotation axis and relative errors of rotation angle versus rotation axis. Translation: (1.732, 0, −1.732). With the horizontal index from 0 to 20, the rotation axis changes from (1, 0, 0) to (0, 0, 1) in *X*-*Z* plane at 20 evenly spaced directions. Rotation angle: (a) 5°; (b) 15°.

nificantly influence those errors. However, as discussed earlier, when the error of rotation axis and rotation angle are considered separately (instead of error in R as a whole), the rotation axis has different effects. This can be seen from Fig. 14, which shows the errors of rotation axis and rotation angle with the same motion parameters as those in Fig. 13. The results do show that when the rotation axis changes from being parallel to the image plane to being orthogonal to image plane, the errors in rotation axis slightly decrease while those in rotation angle slightly increase.

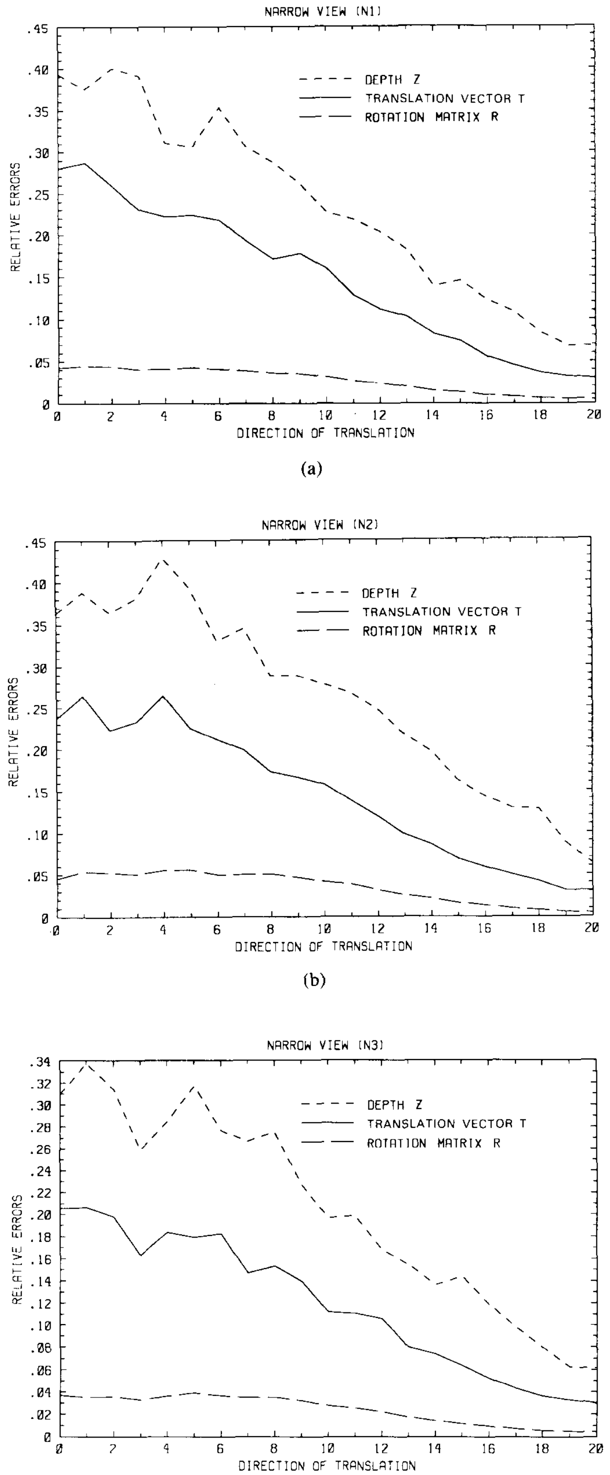
2) System Parameters: To show the effects of decreasing image size (field of view), the image size is reduced by a factor of 2. To make sure that the same scene is visible and cover roughly the same area of the images as earlier, the object is moved away from the camera such that object distance is doubled. Other parameters are kept unchanged. Then with the same motion parameters as in Fig. 11, the errors become those shown in Fig. 15. Comparing those in Fig. 11, the corresponding curves in Fig. 15 increases more drastically as horizontal index decreases and the corresponding errors are much larger. This means that the reduction of image size particularly worsens the cases where the translation direction is nearly parallel to image plane. All of those are again consistent with the discussion in Section IV-C.

Fig. 16 shows the relations between the errors in the estimates and the image resolution. It can be seen that reducing the resolution by a factor of two roughly doubles the errors, which is expected according to statistics.

The simulations shown in this subsection have justified the qualitative observations discussed in Section IV. The numerical examples of the simulations also give a quantitative evaluation of the effects of the parameters.

D. Results for Images of Real World Scenes

The algorithm has been tested on real world images. Fig. 17 shows a pair of images of a scene in our laboratory. A CCD video camera with roughly 512 by 512 resolution is used as imaging sensor. The focal length of the

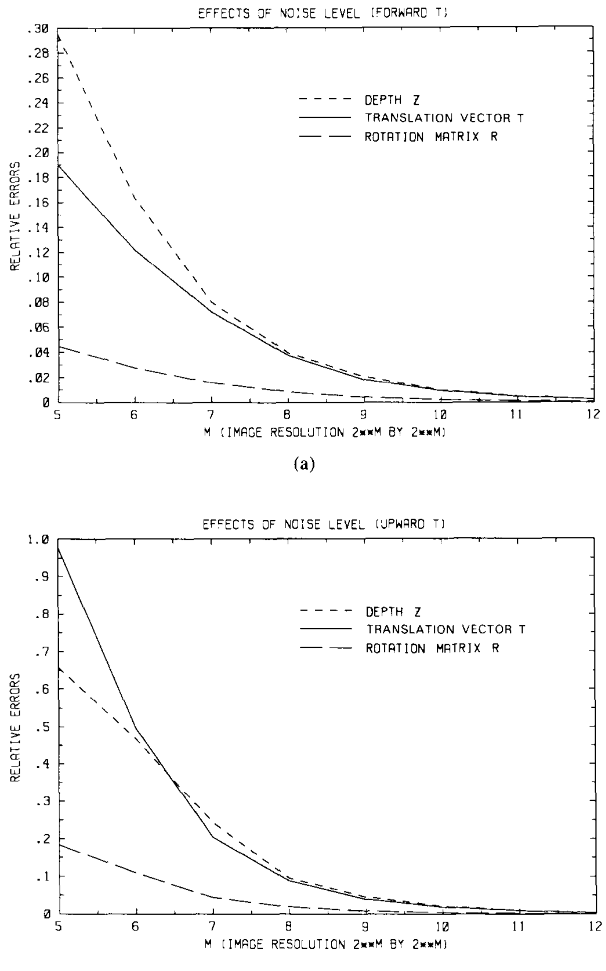


(a)

(b)

(c)

Fig. 15. Relative errors versus direction of translation with smaller image size (same resolution) than that in Fig. 11. Rotation axis: (a) (1, 0, 0); (b) (0, 1, 0); (c) (0, 0, 1). Rotation angle: 5°. With the horizontal index from 0 to 20, the direction of the translation changes from (1, 0, 0) to (0, 0, 1) with evenly spaced 20 directions. The magnitude of translation is fixed to be 3.

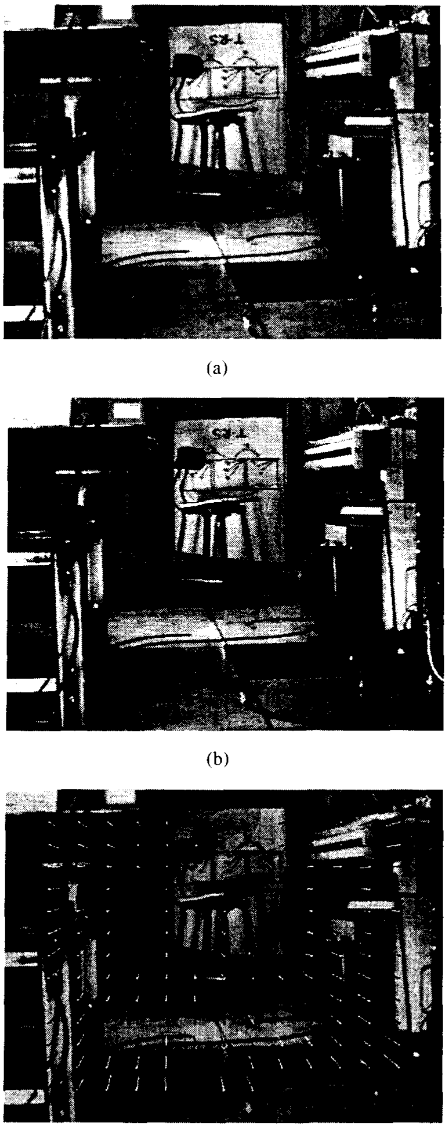


(a)

(b)

Fig. 16. Relative errors versus image resolution. Translation: (a) (0, 0, −3), orthogonal to image plane; (b) (3, 0, 0), parallel to image plane. Rotation axis: (1, 1, 1); Rotation angle: 5°. Image resolution: $2^m \times 2^m$.

camera is simply calibrated but no nonlinear correction is made for the camera. The camera takes two images at different positions. A two-view matcher computes image displacement field on a dense pixel grid [33]. The sample on a 13 by 13 grid of the displacement field computed is shown in Fig. 17(c), overlaid on the first image. Those 13 × 13 = 169 displacement vectors shown in Fig. 17 are used as point correspondences. The motion parameters computered are shown in Table I. Since no attempt has been made to obtain ground truth, we do not know the accuracy of those motion parameters. Instead, we measure the discrepancy between the image of the reconstructed 3-D structure and the computed point correspondences (determined by displacement field). Let d_i be the distance in image between the projection of the reconstructed 3-D point i and that given by the displacement field in the first image, d_i' be that of the second image. We define (*standard*) *image error* as $\sqrt{\sum_{i=1}^{n} (d_i^2 + (d_i')^2)/2n}$, where n is the number of point



(a)

(b)

(c)

Fig. 17. (a), (b) Two views of a scene in a laboratory. (c) Sample of the displacement field computed.

TABLE I
DATA AND RESULTS FOR IMAGES OF A LABORATORY SCENE

Translation	0.052711	−0.060781	0.996758
Rotation axis	0.244922	0.644978	−0.723890
Rotation angle		0.362396°	
Image error		0.000784	
Pixel width		0.000938	

in an image. Image error indicates the discrepancy between the inferred images and the observed images. Since a large number of point correspondences are used, the image error for the pair of images is within a pixel width, which is satisfactory.

VI. CONCLUSIONS

A new algorithm is presented which gives a closed-form solution for the motion parameters and the relative depths of the 3-D points. It exploits the redundancy available in the data to combat noise. It is simpler and more stable than the existing linear algorithms. The error estimation for motion parameters is based on first order perturbation. The results of simulations show a strong correlation between the estimated and the actual errors. The cases of degeneracy or near degeneracy are identified and the reliability of the solution in the presence of noise is indicated.

Based on the qualitative analysis and the results of simulations, the following observations can be made about reliable motion estimation.

1) The focal length of the image sensor should be short (or field of view should be large) to obtain more reliable estimates.

2) The magnitude of translation should be large to allow stable estimation of translation direction and structure of the scene.

3) A translation orthogonal to the image plane allows more stable estimation.

4) The scene should be close to the imaging sensor and the motion should be large to yield long displacement vectors in the image plane.

5) Rotation parameters are not significant to the reliability of the solutions (assuming rotation error is represented by the error of rotation matrix). However, a rotation about an axis orthogonal to the image plane generally results in a slightly more reliable estimate for the rotation axis and less reliable estimate for rotation angle compared with the rotation about an axis parallel to the image plane.

APPENDIX A

Theorem: Let $A = [a_{ij}]$ be an n by n symmetrical matrix and H be an orthonormal matrix such that

$$H^{-1}AH = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_n\}. \tag{A.1}$$

Let the eigenvalues be ordered in nondecreasing order. Without loss of generality, consider the eigenvalue λ_1. Assume λ_1 is a simple eigenvalue.

$$\lambda_1 < \lambda_2 \le \lambda_3 \le \cdots \le \lambda_n. \tag{A.2}$$

Let

$$H = [\boldsymbol{h}_1 \, \boldsymbol{h}_2 \cdots \boldsymbol{h}_n] \tag{A.3}$$

and $\boldsymbol{X}$ be an eigenvector of A associated with λ_1. Let $\boldsymbol{X}(\epsilon)$ be the eigenvector of the perturbed matrix $A(\epsilon) = A + \Delta_A$ associated with the perturbed eigenvalue $\lambda_1(\epsilon)$. $\boldsymbol{X}(\epsilon)$ can be written as

$$\boldsymbol{X}(\epsilon) = \boldsymbol{X} + \boldsymbol{\delta}_x \tag{A.4}$$

with $\boldsymbol{\delta}_x \in \text{span}\{\boldsymbol{h}_2, \boldsymbol{h}_3, \cdots, \boldsymbol{h}_n\}$. Letting ϵ be the maximum absolute value of the elements in $\Delta_A = [\delta_{a_{ij}}]$, we have

$$\Delta_A = \epsilon B \tag{A.5}$$

where $B = [b_{ij}]$, with $b_{ij} = \delta_{a_{ij}}/\epsilon$. Therefore $|b_{ij}| \leq 1$, $1 \leq i \leq n$, $1 \leq j \leq n$. Then for sufficiently small ϵ, the perturbation of λ_1 can be expressed by a convergent series in ϵ:

$$\delta_{\lambda_1} \triangleq \lambda_1(\epsilon) - \lambda_1 = p_1\epsilon + p_2\epsilon^2 + p_3\epsilon^3 + \cdots \qquad (A.6)$$

and the perturbation vector $\boldsymbol{\delta}_x$ can be expressed by a convergent vector series in the space span $\{\boldsymbol{h}_2, \boldsymbol{h}_3, \cdots, \boldsymbol{h}_n\}$. In other words, letting $H_2 = [\boldsymbol{h}_2, \boldsymbol{h}_3, \cdots, \boldsymbol{h}_n]$, then for sufficiently small ϵ, there exist $(n-1)$-dimensional vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \cdots$ such that

$$\boldsymbol{\delta}_x = \epsilon H_2 \boldsymbol{g}_1 + \epsilon^2 H_2 \boldsymbol{g}_2 + \epsilon^3 H_2 \boldsymbol{g}_3 + \cdots . \qquad (A.7)$$

The linear term (in ϵ) in (A.6) is given by

$$p_1\epsilon = \boldsymbol{h}_1^T \Delta_A \boldsymbol{h}_1. \qquad (A.8)$$

The linear term (in ϵ) in (A.7) is given by

$$\epsilon H_2 \boldsymbol{g}_1 = H\Delta H^T \Delta_A \boldsymbol{X} \qquad (A.9)$$

where

$$\Delta = \text{diag}\left\{0, (\lambda_1 - \lambda_2)^{-1}, \cdots, (\lambda_1 - \lambda_n)^{-1}\right\}. \qquad (A.10)$$

That is, suppressing the second and higher order terms (i.e., considering first order perturbation), for the eigenvalue we have

$$\delta_{\lambda_1} \cong \boldsymbol{h}_1^T \Delta_A \boldsymbol{h}_1 \qquad (A.11)$$

and for the eigenvector:

$$\boldsymbol{\delta}_x \cong H\Delta H^T \Delta_A \boldsymbol{X}. \qquad (A.12)$$

Proof: Under the assumption of simple eigenvalue and the definition of ϵ, there exits a positive δ such that if $\epsilon < \delta$, λ_1 is represented by a scalar convergent power series in ϵ with constant term being λ_1:

$$\lambda_1(\epsilon) = \lambda_1 + \epsilon p_1 + \epsilon^2 p_2 + \cdots \qquad (A.13)$$

and the eigenvector of $A(\epsilon)$, $\boldsymbol{X}(\epsilon)$, associated with $\lambda_1(\epsilon)$ is represented by a convergent (vector) power series in ϵ with the constant vector term being $\boldsymbol{X}$:

$$\boldsymbol{X}(\epsilon) = \boldsymbol{X} + \epsilon z_1 + \epsilon^2 z_2 + \cdots \triangleq \boldsymbol{X} + \boldsymbol{\delta}_x \qquad (A.14)$$

(see, for example, [36]). $\boldsymbol{\delta}_x$ can be represented in the basis $\boldsymbol{h}_1, \boldsymbol{h}_2, \cdots, \boldsymbol{h}_n$. The component of $\boldsymbol{\delta}_x$ on $\boldsymbol{h}_1$ can always be included into $\boldsymbol{X}$ since $\boldsymbol{X}$ is any vector collinear with $\boldsymbol{h}_1$. So, $\boldsymbol{\delta}_x \in$ span $\{\boldsymbol{h}_2, \boldsymbol{h}_3, \cdots, \boldsymbol{h}_n\}$. That is, there exist $(n-1)$-dimensional vectors $\boldsymbol{g}_1, \boldsymbol{g}_2, \boldsymbol{g}_3, \cdots$ such that (A.7) holds, or

$$\boldsymbol{X}(\epsilon) = \boldsymbol{X} + \epsilon H_2 \boldsymbol{g}_1 + \epsilon^2 H_2 \boldsymbol{g}_2 + \epsilon^3 H_2 \boldsymbol{g}_3 + \cdots . \qquad (A.15)$$

We have

$$(A + \epsilon B)\boldsymbol{X}(\epsilon) = \lambda_1(\epsilon)\,\boldsymbol{X}(\epsilon). \qquad (A.16)$$

Substituting $\lambda_1(\epsilon)$ using (A.13) and $\boldsymbol{X}(\epsilon)$ using (A.14) we have

$$\begin{aligned}(A + \epsilon B)(\boldsymbol{X} + \epsilon H_2 \boldsymbol{g}_1 + \epsilon^2 H_2 \boldsymbol{g}_2 + \epsilon^3 H_2 \boldsymbol{g}_3 + \cdots)\\ = (\lambda_1 + \epsilon p_1 + \epsilon p_2 + \cdots)\boldsymbol{X}(\epsilon).\end{aligned} \qquad (A.17)$$

(A.17) holds for all $\epsilon < \delta$. Therefore the coefficient vectors of ϵ on the both sides should be equal:

$$AH_2\boldsymbol{g}_1 + B\boldsymbol{X} = \lambda_1 H_2 \boldsymbol{g}_1 + p_1\boldsymbol{X} \qquad (A.18)$$

or, using (A.1):

$$H_2 \,\text{diag}\,\{\lambda_2, \lambda_3, \cdots, \lambda_n\}\boldsymbol{g}_1 - \lambda_1 H_2 \boldsymbol{g}_1 + B\boldsymbol{X} = p_1\boldsymbol{X} \qquad (A.19)$$

Premultiplying both sides by $\boldsymbol{X}^T$ and noticing $\boldsymbol{X}^T H_2 = 0$, we get

$$\boldsymbol{X}^T B\boldsymbol{X} = p_1 \|\boldsymbol{X}\|^2 \qquad (A.20)$$

or

$$p_1 = \boldsymbol{h}_1^T B \boldsymbol{h}_1. \qquad (A.21)$$

Then (A.8) follows immediately. Premultiplying both sides of (A.19) by H_2^T and noticing H is an orthonormal matrix, we get

$$\text{diag}\,\{\lambda_1 - \lambda_2, \lambda_1 - \lambda_3, \cdots, \lambda_1 - \lambda_n\}\boldsymbol{g}_1 = H_2^T B\boldsymbol{X}. \qquad (A.22)$$

We have

$$\begin{aligned}\boldsymbol{g}_1 = \text{diag}\,&\left\{(\lambda_1 - \lambda_2)^{-1}, (\lambda_1 - \lambda_3)^{-1},\right.\\ &\left.\cdots, (\lambda_1 - \lambda_n)^{-1}\right\}H_2^T B\boldsymbol{X}.\end{aligned} \qquad (A.23)$$

The linear term in (A.7) is then

$$\begin{aligned}\epsilon H_2 \boldsymbol{g}_1 &= \epsilon H_2 \,\text{diag}\left\{(\lambda_1 - \lambda_2)^{-1}, (\lambda_1 - \lambda_3)^{-1},\right.\\ &\qquad \left.\cdots, (\lambda_1 - \lambda_n)^{-1}\right\}H_2^T B\boldsymbol{X}\\ &= H\Delta H^T \epsilon B\boldsymbol{X} = H\Delta H^T \Delta_A \boldsymbol{X}.\end{aligned} \qquad (A.24)$$

□

Appendix B

We need to prove (2.35):

$$\|RC - D\|^2 = \boldsymbol{q}^T B\boldsymbol{q} \qquad (B.1)$$

where $C = [\boldsymbol{C}_1\ \boldsymbol{C}_2 \cdots \boldsymbol{C}_n]$, $D = [\boldsymbol{D}_1\ \boldsymbol{D}_2 \cdots \boldsymbol{D}_n]$, R is a rotation matrix and B is defined by

$$B = \sum_{i=1}^{n} B_i^T B_i \qquad (B.2)$$

where

$$B_i = \begin{bmatrix} 0 & (\boldsymbol{C}_i - \boldsymbol{D}_i)^T \\ \boldsymbol{D}_i - \boldsymbol{C}_i & [\boldsymbol{D}_i + \boldsymbol{C}_i]_\times \end{bmatrix}. \qquad (B.3)$$

Rotation matrix R and unit $\boldsymbol{q}$ are related by (2.22).

We first introduce quaternions. A quaternion $\boldsymbol{q}$ consists of a scale component q_0 and a component of a three-dimensional vector $\boldsymbol{Q} = (q_1, q_2, q_3)^T$:

$$\boldsymbol{q} = q_0 + \boldsymbol{Q}. \tag{B.4}$$

Two quaternions are equal if and only if the corresponding components are equal. A vector quaternion is a quaternion with zero scalar component. A scalar quaternion is a quaternion with zero vector component. For convenience, we regard a scalar as a scalar quaternion and a three-dimensional vector as vector quaternion. The conjugate of a quaternion $\boldsymbol{q}$, denoted by $\tilde{\boldsymbol{q}}$, is defined by $\tilde{\boldsymbol{q}} = q_0 - \boldsymbol{Q}$. The addition of two quaternions is defined by

$$(q_0 + \boldsymbol{Q}) + (p_0 + \boldsymbol{P}) = (q_0 + p_0) + (\boldsymbol{Q} + \boldsymbol{P}). \tag{B.5}$$

The multiplication of two quaternions, denoted by "$*$", is defined by

$$\begin{aligned}(q_0 + \boldsymbol{Q}) * (p_0 + \boldsymbol{P}) \\ = (q_0 p_0 - \boldsymbol{Q} \cdot \boldsymbol{P}) + (q_0 \boldsymbol{P} + p_0 \boldsymbol{Q} + \boldsymbol{Q} \times \boldsymbol{P})\end{aligned} \tag{B.6}$$

where "$\cdot$" and "$\times$" are vector dot product and cross product operations, respectively. It is easy to prove that quaternion multiplication is not commutative (unless a quaternion is scalar), but is associative, and is distributive over additions. The conjugate of $\boldsymbol{p} * \boldsymbol{q}$ is equal to $\tilde{\boldsymbol{q}} * \tilde{\boldsymbol{p}}$. Define the norm of a quaternion, $\| \cdot \|$, by a nonnegative value such that $\|\boldsymbol{q}\|^2 = \boldsymbol{q} * \tilde{\boldsymbol{q}}$. It is easy to prove $\boldsymbol{q} * \tilde{\boldsymbol{q}} = \Sigma_{i=0}^{4} q_i^2$. Therefore, the norm of a quaternion is equal to the Euclidean norm of the corresponding four-dimensional space. A quaternion with a unit norm is called unit quaternion. So, for unit quaternion $\boldsymbol{q}$ we have $\boldsymbol{q} * \tilde{\boldsymbol{q}} = 1$.

For a rotation about unit axis $\boldsymbol{n} = (n_x, n_y, n_z)$ by an angle θ, letting $\boldsymbol{q}$ be a unit quaternion such that

$$\boldsymbol{q} = \cos(\theta/2) + \sin(\theta/2)\boldsymbol{n} \tag{B.7}$$

and R be the corresponding rotation matrix,

$$R = \begin{bmatrix} (n_x^2 - 1)(1 - \cos\theta) + 1 & n_x n_y (1 - \cos\theta) - n_z \sin\theta & n_x n_z (1 - \cos\theta) + n_y \sin\theta \\ n_y n_x (1 - \cos\theta) + n_z \sin\theta & (n_y^2 - 1)(1 - \cos\theta) + 1 & n_y n_z (1 - \cos\theta) - n_x \sin\theta \\ n_z n_x (1 - \cos\theta) - n_y \sin\theta & n_z n_y (1 - \cos\theta) + n_x \sin\theta & (n_z^2 - 1)(1 - \cos\theta) + 1 \end{bmatrix} \tag{B.8}$$

we have (see [4], [14])

$$RX = \boldsymbol{q} * X * \tilde{\boldsymbol{q}} \tag{B.9}$$

for any three-dimensional vector $\boldsymbol{X}$. If θ is incremented by 360°, the sign of $\boldsymbol{q}$ in (B.7) is changed but the rotation is not changed. Obviously $-\boldsymbol{q}$ also satisfies (B.9). Therefore $\boldsymbol{q}$ and $-\boldsymbol{q}$ represent the same rotation. In fact, letting Q be the group of unit quaternions under quaternion product, and letting $S = \{1, -1\}$, the factor (or quotient) group Q/S and group $SO(3)$ of rotations in Euclidean space R^3 are isomorphic. Equation (B.7) with S defines an isomorphism.

It is convenient to convert quaternion multiplications to matrix multiplications and regard a quaternion as a four-dimensional column vector when it is operated with matrices.

$$\begin{aligned}\boldsymbol{p} * \boldsymbol{q} &= \begin{bmatrix} p_0 \\ \boldsymbol{P} \end{bmatrix} * \begin{bmatrix} q_0 \\ \boldsymbol{Q} \end{bmatrix} \\ &= \begin{bmatrix} p_0 & -\boldsymbol{P}^T \\ \boldsymbol{P} & p_0 I + [\boldsymbol{P}]_\times \end{bmatrix} \begin{bmatrix} q_0 \\ \boldsymbol{Q} \end{bmatrix} \triangleq [\boldsymbol{p}]_l \boldsymbol{q}\end{aligned} \tag{B.10}$$

where $[\cdot]_l$ is a mapping from a quaternion to a 4 by 4 matrix (l stands for *l*eft multiplication). Similarly, we define the mapping $[\cdot]_r$:

$$\begin{aligned}\boldsymbol{q} * \boldsymbol{p} &= \begin{bmatrix} q_0 \\ \boldsymbol{Q} \end{bmatrix} * \begin{bmatrix} p_0 \\ \boldsymbol{P} \end{bmatrix} \\ &= \begin{bmatrix} p_0 & -\boldsymbol{P}^T \\ \boldsymbol{P} & p_0 I - [\boldsymbol{P}]_\times \end{bmatrix} \begin{bmatrix} q_0 \\ \boldsymbol{Q} \end{bmatrix} \triangleq [\boldsymbol{p}]_r \boldsymbol{q}.\end{aligned} \tag{B.11}$$

Now we are ready to prove (B.1). In the following, vectors are augmented to vector quaternions when they are operated with quaternions. Using (B.9) we have

$$\begin{aligned}\|D - RC\|^2 &= \sum_{i=1}^{n} \|\boldsymbol{D}_i - R\boldsymbol{C}_i\|^2 = \sum_{i=1}^{n} \|\boldsymbol{D}_i - \boldsymbol{q} * \boldsymbol{C}_i * \tilde{\boldsymbol{q}}\|^2 \\ &= \sum_{i=1}^{n} \|(\boldsymbol{D}_i * \boldsymbol{q} - \boldsymbol{q} * \boldsymbol{C}_i) * \tilde{\boldsymbol{q}}\|^2 \\ &= \sum_{i=1}^{n} \|(\boldsymbol{D}_i * \boldsymbol{q} - \boldsymbol{q} * \boldsymbol{C}_i)\|^2\end{aligned} \tag{B.12}$$

since $\tilde{\boldsymbol{q}}$ is a unit quaternion and $\|\boldsymbol{p} * \tilde{\boldsymbol{q}}\|^2 = \boldsymbol{p} * \tilde{\boldsymbol{q}} * \boldsymbol{q} * \tilde{\boldsymbol{p}} = \boldsymbol{p} * \tilde{\boldsymbol{p}} = \|\boldsymbol{p}\|^2$. Using our matrix notation, we have

$$\begin{aligned}\boldsymbol{D}_i * \boldsymbol{q} - \boldsymbol{q} * \boldsymbol{C}_i &= [\boldsymbol{D}_i]_l \boldsymbol{q} - [\boldsymbol{C}_i]_r \boldsymbol{q} \\ &= ([\boldsymbol{D}_i]_l - [\boldsymbol{C}_i]_r)\boldsymbol{q} = B_i \boldsymbol{q}\end{aligned} \tag{B.13}$$

where B_i is readily determined by our mapping (B.10) and (B.11) and it is presented in (B.3). Finally, from (B.12) and (B.13) we have

$$\|D - RC\|^2 = \sum_{i=1}^{n} \|D_i - RC_i\|^2 = \sum_{i=1}^{n} \|B_i q\|^2$$
$$= \sum_{i=1}^{n} q^T B_i^T B_i q = q^T \left(\sum_{i=1}^{n} B_i^T B_i \right) q \triangleq q^T B q. \qquad \text{(B.12)}$$

In the algorithm, we have $n = 3$. The readers are referred to [4], [14], [27], [11] for related discussions.

Appendix C

If we use (2.16) to solve R, using (2.21) and (3.40), we have

$$\delta_B \cong G_B \delta_K \cong G_B D_K \delta_E \triangleq D_B \delta_E$$

where

$$G_B = 2 \begin{bmatrix} F_{tl} & F_{tr} \\ F_{bl} & F_{br} \end{bmatrix}$$

where (note $T_s = (s_1, s_2, s_3)^T$)

$$F_{tl} = \begin{bmatrix} 2s_1 + e_{23} - e_{32} & 2s_2 + e_{31} - e_{13} & 2s_3 + e_{12} - e_{21} & e_{11} & e_{21} - s_3 \\ -e_{22} - e_{33} & e_{12} & e_{13} & 0 & 0 \\ e_{21} & e_{11} - e_{33} & e_{23} & -s_2 & t_1 \\ e_{31} & e_{32} & -e_{11} - e_{22} & -s_3 & 0 \\ -e_{22} - e_{33} & e_{12} & e_{13} & 0 & 0 \\ 2s_1 + e_{32} - e_{23} & 2s_2 + e_{13} - 3e_{31} & 2s_3 + 3e_{21} - e_{12} & e_{11} & e_{21} + 3s_3 \\ e_{31} & -e_{32} & e_{22} - e_{11} & -s_3 & 0 \\ -e_{21} & e_{11} - e_{33} & e_{23} & s_2 & -s_1 \end{bmatrix}$$

$$F_{tr} = \begin{bmatrix} e_{31} + s_2 & e_{12} + s_3 & e_{22} & e_{32} - s_1 & e_{13} - s_2 & e_{23} + s_1 & e_{33} \\ 0 & s_2 & -s_1 & 0 & s_3 & 0 & -s_1 \\ 0 & 0 & 0 & 0 & 0 & s_3 & -s_2 \\ s_1 & 0 & -s_3 & s_2 & 0 & 0 & 0 \\ 0 & s_2 & -s_1 & 0 & s_3 & 0 & -s_1 \\ e_{31} - 3s_2 & e_{12} - s_3 & e_{22} & e_{32} + s_1 & e_{13} + s_2 & e_{23} - s_1 & e_{33} \\ s_1 & 0 & s_3 & -s_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s_3 & -s_2 \end{bmatrix}$$

$$F_{bl} = \begin{bmatrix} e_{21} & -e_{11} - e_{33} & e_{23} & -s_2 & s_1 \\ e_{31} & -e_{32} & e_{22} - e_{11} & -s_3 & 0 \\ 2s_1 + 3e_{32} - e_{23} & 2s_2 + e_{13} - e_{31} & 2s_3 + e_{21} - 3e_{12} & e_{11} & e_{21} + s_3 \\ e_{33} - e_{22} & e_{12} & -e_{13} & 0 & 0 \\ e_{31} & e_{32} & -e_{11} - e_{22} & -s_3 & 0 \\ -e_{21} & e_{11} - e_{33} & e_{23} & s_2 & -s_1 \\ e_{33} - e_{22} & e_{12} & -e_{13} & 0 & 0 \\ 2s_1 + e_{32} - 3e_{23} & 2s_2 + 3e_{13} - e_{31} & 2s_3 + e_{21} - e_{12} & e_{11} & e_{21} + s_3 \end{bmatrix}$$

$$F_{br} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & s_3 & -s_2 \\ s_1 & 0 & s_3 & -s_2 & 0 & 0 & 0 \\ e_{31} - s_2 & e_{12} - 3s_3 & e_{22} & e_{32} + 3s_1 & e_{13} + s_2 & e_{23} - s_1 & e_{33} \\ 0 & s_2 & -s_1 & 0 & -s_3 & 0 & s_1 \\ s_1 & 0 & -s_3 & s_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & s_3 & -s_2 \\ 0 & s_2 & -s_1 & 0 & -s_3 & 0 & s_1 \\ e_{31} - s_2 & e_{12} - s_3 & e_{22} & e_{32} + s_1 & e_{13} + 3s_2 & e_{23} - 3s_1 & e_{33} \end{bmatrix}.$$

Acknowledgment

The authors would like to thank X. Zhuang for some illuminating discussions.

References

[1] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Amer. A*, vol. 2, no. 2, pp. 284-299, 1985.

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

[3] S. T. Barnard and W. B. Thompson, "Disparity analysis of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-2, pp. 333-340, July 1980.

[4] O. Bottema and B. Roth, *Theoretical Kinematics*. New York: North-Holland, 1979.

[5] T. J. Broida and R. Chellappa, "Kinematics and structure of a rigid object from a sequence of noisy images: A batch approach," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Miami Beach, FL, June 1986, pp. 176-182.

[6] J. Burns, A. Hanson, and E. Riseman, "Extracting straight lines," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 4, pp. 425-445, July 1986.

[7] L. Davis, Z. Wu, and H. Sun, "Contour based motion estimation," *Computer Vision, Graphics, Imaging Processing*, vol. 23, pp. 313-326, 1983.

[8] L. Dreschler and H.-H. Nagel, "Volumetric model and 3-D trajectory of a moving car derived from monocular TV frame sequences of a street scene," *Comput. Graphics Image Processing*, vol. 20, pp. 199-228, 1982.

[9] J. Q. Fang and T. S. Huang, "Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 547-554, 1984.

[10] 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.

[11] O. D. Faugeras and M. Hebert, "A 3-D recognition and positioning algorithm using geometrical matching between primitive surfaces," in *Proc. 8th Int. Joint Conf. Artificial Intell.*, Karlsruhe, West Germany, Aug. 1983, pp. 996-1002.

[12] F. Glazer, R. Reynolds, and P. Anandan, "Scene matching by hierarchical correlation," in *Proc. IEEE Conf. Computer Vision Pattern Recognition*, June 1983, pp. 432-441.

[13] W. K. Gu, J. Y. Yang, and T. S. Huang, "Matching perspective views of a polyhedron using circuits," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 3, pp. 390-400, May 1987.

[14] W. R. Hamilton, *Elements of Quaternions*, 3rd ed. New York: Chelsea, 1969.

[15] D. J. Heeger, "Optical flow from spatiotemporal filters," in *Proc. Int. Conf. Computer Vision*, London, England, June 1987, pp. 181-190.

[16] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185-203, 1981.

[17] T. S. Huang, "Three-dimensional motion analysis by direct matching," in *Proc. Topical Meeting Machine Vision*, Opt. Soc. America, Lake Tahoe, NV, Mar. 1985.

[18] 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. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 2, pp. 229-244, 1987.

[19] L. Kitchen and A. Rosenfeld, "Gray-level corner detection," *Pattern Recognition Lett.*, vol. 1, pp. 95-102, 1982.

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

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

[22] D. Marr and T. Poggio, "A computational theory of human stereo vision," *Proc. Roy. Soc. London*, Series B, vol. 204, pp. 301-308, 1979.

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

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

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

[26] H. Shariat and K. Price, "Results of motion estimation with more than two frames," in *Proc. Image Understanding Workshop*, Los Angeles, CA, Feb. 23-25, 1987, pp. 694-703.

[27] M. D. Shuster, "Approximate algorithms for fast optimal attitude computation," in *Proc. AIAA Guidance and Control Specialist Conf.*, Palo Alto, CA, Aug. 1978, pp. 88-95.

[28] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of 3-D motion parameters of rigid bodies with curved surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 13-27, 1984.

[29] A. Verri and T. Poggio, "Against quantitative optical flow," in *Proc. First Int. Conf. Computer Vision*, London, England, June 1987, pp. 171-180.

[30] J. Weng, T. S. Huang, and N. Ahuja, "3-D motion estimation, understanding and prediction from noisy image sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, no. 3, pp. 370-389, 1987.

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

[32] J. Weng, T. S. Huang, and N. Ahuja, "Determining motion/structure from line correspondences: A robust algorithm and uniqueness theorems," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Ann Arbor, MI, June 5-9, 1988.

[33] J. Weng, N. Ahuja, and T. S. Huang, "Two-view matching," in *Proc. 2nd Int. Conf. Computer Vision*, Florida, Dec. 1988, pp. 64–73.

[34] Y. Yasumoto and G. Medioni, "Robust estimation of three-dimensional motion parameters from sequence of image frames using regularization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, no. 4, pp. 464–471, 1986.

[35] X. Zhuang, T. S. Huang, and R. M. Haralick, "Two-view motion analysis: A unified algorithm," *J. Opt. Soc. Amer. A*, vol. 3, no. 9, pp. 1492–1500, Sept. 1986.

[36] J. H. Wilkinson, *The Algebraic Eigenvalue Problem.* London: Oxford University Press, 1965.

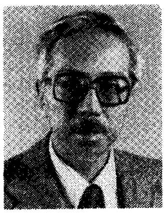
[37] O. A. Zuniga and R. M. Haralick, "Corner detection using the facet model," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 1983, pp. 30–37.

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

Juyang Weng (S'85) was born in Shanghai, China, on April 15, 1957. He received the B.S. degree in computer science from Fudan University, Shanghai, China, in 1982, and the M.S. degree in computer science from the University of Illinois, Urbana-Champaign, in 1985.

Since September 1984 he has been 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. He is now working toward the Ph.D. degree in computer science. His current research interests are in computer vision, image processing, solid modeling and representation, and artificial intelligence.

Mr. Weng is a member of Phi Kappa Phi and Sigma Xi.



Thomas S. Huang (S'61–M'63–SM'76–F'79) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, China, and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge.

He was on the Faculty of the Department of Electrical Engineering at M.I.T. from 1963 to 1973, and on the Faculty of the School of Electrical Engineering and Director of its Laboratory for Information and Signal Processing at Purdue University from 1973 to 1980. In 1980, he joined the University of Illinois at Urbana-Champaign, where he is now Professor of Electrical and Computer Engineering and Research Professor at the Coordinated Science Laboratory. During his sabattical leaves, he has worked at M.I.T. Lincoln Laboratory, IBM Thomas J. Watson Research Center, and the Rheinishes Landes Museum in Bonn, West Germany, and held visiting Professor positions at the Swiss Institutes of Technology in Zürich and Lausanne, the University of Hannover in West Germany, and INRS-Telecommunications of the University of Quebec in Montreal, Canada. He has served as as consultant to numerous industrial firms and government agencies both in the U.S. 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), an A. V. Humboldt Foundation Senior U.S. Scientist Award (1976–1977), and a Fellowship from the Japan Society for the Promotion of Science (1986). He is an Editor of the international journal *Computer Vision, Graphics, and Image Processing*, Editor of the *Springer Series in Information Sciences*, published by Springer-Verlag, and Editor of the *Research Annual Series on Advances in Computer Vision and Image Processing* published by JAI Press.

Narendra Ahuja (S'79–M'79–SM'85) received the B.E. degree with honors in electronics 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, College Park, as a Graduate Research Assistant (1975–1978), as a Faculty Research Assistant (1978–1979), and as a Research Associate (1979). Since 1979 he has been with the University of Illinois as an Assistant Professor (1979–1984), Associate Professor (1984–1988), and Professor (1988–) in the Department of Electrical and Computer Engineering and the Coordinated Science Laboratory. His research interests are in computer vision, robotics, image processing, and parallel algorithms.

Dr. Ahuja received the University Scholar Award (1985) and a Presidential Young Investigator Award (1984). He has coauthored the book *Pattern Models* (Wiley, 1983). He was Program Chair for the 1987 IEEE Workshop on Computer Vision. He is an Associate Editor of *Computer Vision, Graphics and Image Processing*, and a member of the Association of Computing Machinery and the American Association for Artificial Intelligence.