Motion and Structure from Line Correspondences: Closed-Form Solution, Uniqueness, and Optimization

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

Abstract—**This paper discusses estimating motion and structure parameters from line correspondences of a rigid scene. We present in this paper a new closed-form solution to motion and structure parameters from line correspondences through three monocular perspective views. The algorithm makes use of redundancy in the data to improve the accuracy of the solutions. The uniqueness of the solution is established, and necessary and sufficient conditions for degenerate spatial line configurations have been derived. Optimization has been employed to further improve the accuracy of the estimates in the presence of noise. Simulations have showed that the errors of the optimized estimates are close to the theoretical lower error bound.**

Index Terms—**Computer vision, dynamic scene analysis, motion estimation, optimal estimation, structure from motion.**

I. Introduction

From monocular image sequences taken by a camera undergoing motion relative to a scene, one generally can determine the parameters of motion and structure of the scene up to a scale factor. Conceptually, three steps may be involved in motion and structure analysis, although practically, a merge of these steps is possible. First, features are extracted from images. Then, interframe correspondences between the selected features are established. Finally, the motion and structure are computed from these feature correspondences. This paper is devoted mainly to the last step.

In order to focus on the major problems to be discussed, we assume that the scene is rigid. This assumption is valid for the cases where a camera moves in a static scene. If the scene consists of individually moving rigid objects, the images need to be segmented into regions where each corresponds to a rigid part of the scene. This can be done by, e.g., segmenting displacement field based on the rigidity [7], [1]. The results of this paper can then be applied to each of those rigid regions. Such a segmentation is beyond the scope of this paper. It is worth noting that a relatively wide field of view is very crucial to the reliability of solutions in the presence of noise [23], [28].

Object points are a type of feature commonly used by feature-based algorithms. The closed-form solutions to motion and structure of the scene from feature points are available [14], [22], [25], and the condition of the corresponding degenerate spatial configurations is known [15]. In the category of optical flow-based methods, several methods for the corresponding problem have been developed [30], [31], [24].

The choice of types of features depends on their availability in the images and the reliability of their measurement. When points are not available in large quantities, other features such as lines or contours can be used [29], [12], [9], [6]. Since higher level features like lines, edges, and contours are determined by a set of pixels, the redundancy in the edge pixels make it possible to locate those features accurately in image plane. In this paper, we discuss motion estimation based on lines. In practice, one certainly may use different types of features to obtain robust solutions. However, the study of the use of single type of feature is very important, theoretically and practically, to the general use of multiple types of features. It provides insights into the roles of this type of feature in the solutions that use multiple feature types. Our general approach to using lines might also be useful to the use of other types of features. The optimization to be discussed in this paper also allows the use of multiple types of features.

The lines used in this study are straight lines without known end points since the end points of an extracted line are very unstable [4], [9], [16]. For example, the end points often do not correspond to physical points and move as the view point changes. Many factors such as lighting and surface reflection often change the position of the end points when the view point changes. However, the location and orientation of the line can generally be determined reliably by a line fitting along a sequence of edge points. In other words, long lines are preferred since they provide more edge points to allow a more accurate measurement of the line position.

From line correspondences through three perspective views, Yen and Huang [29] and Liu and Huang [12] iteratively solve a set of nonlinear equations for the motion parameters. A different approach is reported by Mitiche *et al.* [17], where the property of angular invariance between lines is used. Faugeras *et al.* [9] approximate the nonlinear equations by linear equations and use iterated extended Kalman filter to estimate the motion parameters (the filter is also a nonlinear iterative search method). These algorithms do not give a closed-form solution to the problem.

Spetsakis and Aloimonos [21], and Liu and Huang [13] recently developed linear algorithms for estimating motion and structure parameters from line correspondences. The basic strategies are similar to those of point-based linear algorithms. First, a set of intermediate parameters are estimated by solving linear equations. Then, the motion parameters are solved from those intermediate parameters. Although closed-form solutions

Manuscript received May 28, 1989; revised September 21, 1991. This work was supported by the National Science Foundation under grants ECS-83-52408 and IRI-8605400. Recommended for acceptance by R. Woodham.

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

IEEE Log Number 9104990.

0162-8828/92$03.00 © 1991 IEEE

are derived in those linear algorithms, many problems remain to be solved. First, many spurious solutions are generated by their algorithms. The number of spurious solutions is so large that the computation is inefficient, and the correct solutions are difficult to identify in the presence of noise. The formulations of the algorithms also generate some degenerate cases that may otherwise be avoided. Second, the question of uniqueness was left unanswered. Do the algorithms give a unique solution? What are the necessary and sufficient conditions for the algorithm to give a unique solution? Third, preliminary experiments have shown that the algorithms are extremely sensitive to noise. Only results for noise-free data were published. These problems are taken up in this paper. In addition, we develop methods to obtain optimal solutions from noise-corrupted lines. We also study the inherent stability of using lines for motion analysis.

We first present a guideline of our approach to those problems. A common characteristic of linear algorithms is solving for a matrix of intermediate unknowns through linear equations. Those intermediate unknowns are not independent. In other words, there exist constraints on the variables of intermediate unknowns, and there are more intermediate unknown variables than the "independent" unknowns. The linear equations are solved without using those constraints (otherwise, we are forced to solve nonlinear equations). The resulting intermediate unknowns contain redundant information. One of the objectives of our linear algorithm is to make good use of such redundancy to improve the accuracy of the solutions in the presence of noise. On the other hand, the problem to be investigated here involves three image frames with line features, and therefore, it is significantly more complicated than a two-frame point-based problem. We derive compact computational steps to avoid, as much as possible, degenerate cases and spurious solutions that may otherwise be generated. As a result, we are able to investigate the uniqueness of solutions. Although our linear algorithm is designed to well utilize the redundancy in the data, the solutions are not overall optimal in the presence of noise. However, those solutions can be used as an initial guess for further improvement through optimization. Since the optimization is nonlinear, the good initial guess provided by the solution of the linear algorithm is very crucial to the correct convergence of the optimization. Finally, we will compare the error variance of our optimal solution with that of a theoretical lower bound.

The remainder of this paper is organized as follows. The linear algorithm is derived in the next section. Section III is devoted to the problem of degeneracy and uniqueness. Optimization is discussed in Section IV. Simulation results are presented in Section V. Section VI presents concluding remarks.

II. Solution and Algorithm

This section presents a linear algorithm for motion and structure estimation. The goal is to determine the relative motion between the camera and the scene, as well as the structure of the scene.

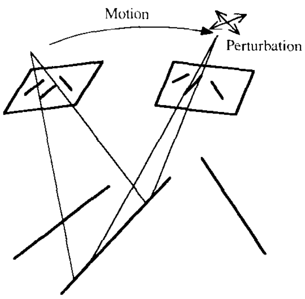


Fig. 1. Motion and structure cannot be determined from lines in two images.

A. Why Two Views Are Not Sufficient

We first show that inherently, motion cannot be determined from lines in just two images. To do this, it is more convenient to consider the situation where a camera is moving in a static scene. Let a camera system consist of a projection center and an image plane. At each time instant, define a camera system at the corresponding position along the trajectory of motion associated with the corresponding image. The problem to be investigated is equivalent to the following: Fixing the first camera system at a known position and orientation, we want to determine the position and orientation of the second camera system and the 3-D positions of the lines from the projections of lines in the two image planes (see Fig. 1). We show that it is impossible. For each line in an image plane, we define a 3-D plane called the projection plane of the line, which passes through the projection center and the line. For each line correspondence, two camera systems determine two corresponding projection planes, whose intersection gives the line in 3-D. Suppose that the second camera system is *arbitrarily* perturbed away from the correct position with the projection planes fixed with the camera system. Since any two nonparallel 3-D planes intersect and the corresponding intersection yields a line, every pair of projection planes still intersects as long as the perturbation is not so large that two corresponding projection planes become parallel (see Fig. 1). In other words, the arbitrarily perturbed position of the second camera system still gives a 3-D line configuration that is consistent with the two images observed. Therefore, the solution to the position and orientation of the camera is arbitrary (at least in an open set including the correct one), as is the corresponding 3-D line structure. If a third image is added, it is possible to determine the position of the second and the third camera systems as well as the 3-D position of the lines because the intersection of three projection planes is generally not a line. In the following, we discuss determining motion and 3-D line structure using line correspondences through three images.

B. Three Views

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 toward the scene. Therefore, in this coordinate system, the camera is fixed, and the scene is moving. By the pin-hole camera model, the image and the focal length can be scaled by any positive common factor without changing the direction of the projection lines. If we scale the image and focal length by the reciprocal of the focal length, we obtain a *normalized camera model,* in which the focal length is one, and the normalized image size determines the field of view. Therefore, without loss of generality, we consider the normalized camera model. Visible objects are always located in front of the camera, i.e., $z > 0$. ($z \leq 1$ can occur since the focal length is normalized.)

We introduce some notation to simplify the presentation. A vector is regarded as both vector and column matrix. Therefore, vector operations and matrix operations can be applied to 3-D vectors with matrix operations taking precedence over vector operations. $//$ denotes "parallel" relationship. $\boldsymbol{a}//\boldsymbol{b}$ if and only if $\boldsymbol{a} \times \boldsymbol{b} = \boldsymbol{0}$. For a matrix $A = [a_{ij}]$, $\|\cdot\|$ denotes the Euclidean norm: $\|[a_{ij}]\|^2 = \sum_{ij} a_{ij}^2$. $[X]_\times$ is a 3×3 skew symmetric matrix determined by 3-D vector $\boldsymbol{X}$ such that $\boldsymbol{X} \times \boldsymbol{Y} = [\boldsymbol{X}]_\times \boldsymbol{Y}$ holds for any 3-D vector $\boldsymbol{Y}$ [28].

Consider a coordinate system fixed on the camera. A line passing through a point $\boldsymbol{x}_p$ (to be specific, let $\boldsymbol{x}_p$ be the point on the line that is the closest to the origin) with direction $\boldsymbol{l}$ at time t_0 can be expressed in the following parametric form:

$$\boldsymbol{x}_0 = \boldsymbol{x}_p + k\boldsymbol{l}$$

where the subscript in $\boldsymbol{x}_0$ means time t_0, and k is the parameter. At another time instant t_1, the line is moved from t_0 by a rotation represented by a rotation matrix $\boldsymbol{R}$ and a translation represented by a translation vector $\boldsymbol{T}$, that is, any point at position $\boldsymbol{x}_1$ at time t_1 is related to its position $\boldsymbol{x}_0$ at times t_0 by

$$\boldsymbol{x}_1 = R\boldsymbol{x}_0 + \boldsymbol{T}. \tag{2.1}$$

The line equation at time t_1 is

$$t_1: \quad \boldsymbol{x}_1 = R\boldsymbol{x}_0 + \boldsymbol{T} = (R\boldsymbol{x}_p + \boldsymbol{T}) + kR\boldsymbol{l}. \tag{2.2}$$

It is easy to see that after motion, the line at time t_1 passes through point $R\boldsymbol{x}_p + \boldsymbol{T}$ with a direction $R\boldsymbol{l}$. Similarly, at another time instant t_2, the line is rotated by a rotation matrix S and then translated by a vector $\boldsymbol{U}$ from time t_0. The line equation at t_2 is

$$t_2: \quad \boldsymbol{x}_2 = S\boldsymbol{x}_0 + \boldsymbol{U} = (S\boldsymbol{x}_p + \boldsymbol{U}) + kS\boldsymbol{l}. \tag{2.3}$$

Notice that it is not necessary that $t_0 \leq t_1 \leq t_2$ holds. The order of the three time instants t_0, t_1, and t_2 can be arbitrary (see Fig. 2).

C. Two Important Equations

We define the *projection plane* of a line as the plane that passes through the line and the projection center and define *projection normal* of a line as the normal of the projection plane. Since the projection normal of a line is orthogonal to the line and the position vector of any point on the line, it is easy to get the projection normal at the three time instants from (2.1)–(2.3):

$$t_0: \quad \boldsymbol{n}_0 = \boldsymbol{x}_p \times \boldsymbol{l} \tag{2.4}$$

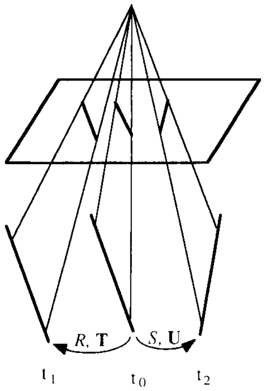


Fig. 2. Motion and structure from lines in three images.

$$\begin{aligned} t_1: \quad \boldsymbol{n}_1 &= (R\boldsymbol{x}_p + \boldsymbol{T}) \times R\boldsymbol{l} = R\big((\boldsymbol{x}_p + R^{-1}\boldsymbol{T}) \times \boldsymbol{l}\big) \\ &= R\big(\boldsymbol{n}_0 + R^{-1}\boldsymbol{T} \times \boldsymbol{l}\big) \end{aligned} \tag{2.5}$$

$$\begin{aligned} t_2: \quad \boldsymbol{n}_2 &= (S\boldsymbol{x}_p + \boldsymbol{U}) \times S\boldsymbol{l} = S\big((\boldsymbol{x}_p + S^{-1}\boldsymbol{U}) \times \boldsymbol{l}\big) \\ &= S\big(\boldsymbol{n}_0 + S^{-1}\boldsymbol{U} \times \boldsymbol{l}\big). \end{aligned} \tag{2.6}$$

Equation (2.5) gives

$$R^{-1}\boldsymbol{n}_1 = \boldsymbol{n}_0 + R^{-1}\boldsymbol{T} \times \boldsymbol{l}. \tag{2.7}$$

Using the vector identity $\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{a} \cdot \boldsymbol{c})\boldsymbol{b} - (\boldsymbol{a} \cdot \boldsymbol{b})\boldsymbol{c}$ and (2.7) yields

$$\begin{aligned} \boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1 &= \boldsymbol{n}_0 \times \big(R^{-1}\boldsymbol{T} \times \boldsymbol{l}\big) \\ &= (\boldsymbol{n}_0 \cdot \boldsymbol{l})R^{-1}\boldsymbol{T} - \big(\boldsymbol{n}_0 \cdot R^{-1}\boldsymbol{T}\big)\boldsymbol{l} \\ &= -\big(\boldsymbol{n}_0 \cdot R^{-1}\boldsymbol{T}\big)\boldsymbol{l}. \end{aligned} \tag{2.8}$$

The last equation follows from the fact that $\boldsymbol{n}_0 \cdot \boldsymbol{l} = 0$. Using $\boldsymbol{n}_0$ solved from (2.7) gives

$$\begin{aligned} \boldsymbol{n}_0 \cdot R^{-1}\boldsymbol{T} &= \big(R^{-1}\boldsymbol{n}_1 - R^{-1}\boldsymbol{T} \times \boldsymbol{l}\big) \cdot R^{-1}\boldsymbol{T} \\ &= R^{-1}\boldsymbol{n}_1 \cdot R^{-1}\boldsymbol{T} = \boldsymbol{n}_1 \cdot \boldsymbol{T}. \end{aligned} \tag{2.9}$$

Equations (2.8) and (2.9) yield

$$\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1 = -(\boldsymbol{n}_1 \cdot \boldsymbol{T})\boldsymbol{l}. \tag{2.10}$$

Similarly, we get

$$\boldsymbol{n}_0 \times S^{-1}\boldsymbol{n}_2 = -(\boldsymbol{n}_2 \cdot \boldsymbol{U})\boldsymbol{l}. \tag{2.11}$$

D. A Geometrical View

These two equations, (2.10) and (2.11), can also be proved geometrically as shown in the following: For each line, we arbitrarily choose its direction from two possible alternative ones and thus represent it by a vector $\boldsymbol{l}$. Viewed along this direction, the configuration can be shown in Fig. 3, where the line vector $\boldsymbol{l}$ points towards the paper (a cross marks the tail of the vector), and the point "O" denotes the projection center of the camera. We first assume that the line does not go through the projection center "O." From the original motion equation (2.1), we have

$$R^{-1}\boldsymbol{x}_1 = \boldsymbol{x}_0 + R^{-1}\boldsymbol{T}.$$

This means that if the moved line is rotated back by R^{-1}, the resulting composite motion is a pure translation represented by vector $R^{-1}\boldsymbol{T}$. This composite motion is shown in Fig. 3 (only the projection of $R^{-1}\boldsymbol{T}$ onto the viewing plane can be shown in the figure). Viewed from the direction employed for Fig. 3, the vector $R^{-1}\boldsymbol{T}$ can lie on either side of the projection plane of the line ($\boldsymbol{l}$ the plane that passes "O" and $\boldsymbol{l}$, which is visible as a line in Fig. 3). In order to show both cases, we let the corresponding vectors $R^{-1}\boldsymbol{T}$ (first motion) and $S^{-1}\boldsymbol{U}$ (second motion) lie on different sides in Fig. 3. The vector $R^{-1}\boldsymbol{n}_1$ is the projection normal of the plane after the pure translation $R^{-1}\boldsymbol{T}$, and therefore, it is orthogonal to the line that is translated by vector $R^{-1}\boldsymbol{T}$ from $\boldsymbol{l}$. Because the composite motion is a translation, the vector $R^{-1}\boldsymbol{n}_1$ is also orthogonal to $\boldsymbol{l}$ as shown in Fig. 3. Therefore, both $R^{-1}\boldsymbol{n}_1$ and $\boldsymbol{n}_0$ are orthogonal to $\boldsymbol{l} : \boldsymbol{l}//(\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1)$, which gives the "alignment part" of (2.10). By definition in (2.4), the length of $\boldsymbol{n}_0$, d_0, is equal to the distance between the line and the origin. In Fig. 3, there exist two congruent right triangles determined by two equal angles β's and the two equal hypotenuses with length d_0. The corresponding sides opposite to β's, respectively, should be equal: One side is equal to $\|\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1\|/\|\boldsymbol{n}_1\|$, and the other is equal to $R^{-1}\boldsymbol{n}_1 \cdot R^{-1}\boldsymbol{T}/\|\boldsymbol{n}_1\| = \boldsymbol{n}_1\boldsymbol{T}/\|\boldsymbol{n}_1\|$. This proves the "length part" of (2.10). What remains to be established is the "sign part." As we mentioned above, the vector $R^{-1}\boldsymbol{T}$ can lie on either side of the projection plane of $\boldsymbol{l}$: the side as in Fig. 3 or the side of $S^{-1}\boldsymbol{U}$ shown in Fig. 3. In the former case, $\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1$ has a direction opposite to $\boldsymbol{l}$, and we have $\boldsymbol{n}_1 \cdot \boldsymbol{T} \geq 0$ because $R^{-1}\boldsymbol{n}_1$ and $R^{-1}\boldsymbol{T}$ lie on the same side of the projection plane of $\boldsymbol{l}$ and the angle between them is an internal angle of the right triangle. $\boldsymbol{n}_1 \cdot \boldsymbol{T} = 0$ holds true if and only if $\boldsymbol{T} = \boldsymbol{0}$, and so does $\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1 = \boldsymbol{0}$. This concludes the "sign part" of (2.10) for the former case. For the latter case, the vector $\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1$ gets the same direction as $\boldsymbol{l}$, and $\boldsymbol{n}_1 \cdot \boldsymbol{T} \leq 0$ because $R^{-1}\boldsymbol{n}_1$ and $R^{-1}\boldsymbol{T}$ are located on the different sides of the projection plane of $\boldsymbol{l}$, and the angle between them is an external angle of a right triangle. Therefore, the "sign part" of (2.10) is always true for both cases. Suppose the line $\boldsymbol{l}$ does go through the projection center "O." Then, $\boldsymbol{n}_0 = \boldsymbol{0}$ according to the definition of (2.4), and $R^{-1}\boldsymbol{T}$ is orthogonal to $R^{-1}\boldsymbol{n}_1$ as can be seen from Fig. 3. Thus, (2.10) still holds true since both sides vanish. This completes the proof for (2.10). The proof for (2.11) is analogous. Compared with the geometrical proof, the algebraic derivation discussed earlier appears to be more rigorous but less intuitive. From the geometrical proof, one can see what properties are used to determine the solution.

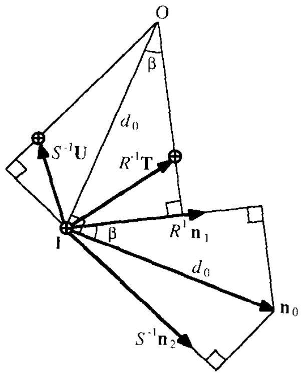


Fig. 3. Geometrical illustration of (2.10) and (2.11).

E. Intermediate Parameters

Multiplying both sides of (2.10) by $\boldsymbol{n}_2 \cdot \boldsymbol{U}$ and those of (2.11) by $\boldsymbol{n}_1 \cdot \boldsymbol{T}$ yields

$$(\boldsymbol{n}_2 \cdot \boldsymbol{U})(\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1) = (\boldsymbol{n}_1 \cdot \boldsymbol{T})(\boldsymbol{n}_0 \times S^{-1}\boldsymbol{n}_2)$$

or

$$[\boldsymbol{n}_0]_\times \boldsymbol{B} = \boldsymbol{0}$$

where $\boldsymbol{B} = (\boldsymbol{n}_2 \cdot \boldsymbol{U})R^{-1}\boldsymbol{n}_1 - (\boldsymbol{n}_1 \cdot \boldsymbol{T})S^{-1}\boldsymbol{n}_2$. Letting $R = [\boldsymbol{R}_1\ \boldsymbol{R}_2\ \boldsymbol{R}_3]$ and $S = [\boldsymbol{S}_1\ \boldsymbol{S}_2\ \boldsymbol{S}_3]$, $\boldsymbol{B}$ can be expressed as

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{n}_1^T(\boldsymbol{R}_1\boldsymbol{U}^T - \boldsymbol{T}\boldsymbol{S}_1^T)\boldsymbol{n}_2 \\ \boldsymbol{n}_1^T(\boldsymbol{R}_2\boldsymbol{U}^T - \boldsymbol{T}\boldsymbol{S}_2^T)\boldsymbol{n}_2 \\ \boldsymbol{n}_1^T(\boldsymbol{R}_3\boldsymbol{U}^T - \boldsymbol{T}\boldsymbol{S}_3^T)\boldsymbol{n}_2 \end{bmatrix} \triangleq \begin{bmatrix} \boldsymbol{n}_1^T E\boldsymbol{n}_2 \\ \boldsymbol{n}_1^T F\boldsymbol{n}_2 \\ \boldsymbol{n}_1^T G\boldsymbol{n}_2 \end{bmatrix}$$

where we define the *intermediate parameters* (E, F, G):

$$E = \boldsymbol{R}_1\boldsymbol{U}^T - \boldsymbol{T}\boldsymbol{S}_1^T, \qquad F = \boldsymbol{R}_2\boldsymbol{U}^T - \boldsymbol{T}\boldsymbol{S}_2^T, \qquad G = \boldsymbol{R}_3\boldsymbol{U}^T - \boldsymbol{T}\boldsymbol{S}_3^T. \tag{2.12}$$

We have

$$[\boldsymbol{n}_0]_\times \begin{bmatrix} \boldsymbol{n}_1^T E\boldsymbol{n}_2 \\ \boldsymbol{n}_1^T F\boldsymbol{n}_2 \\ \boldsymbol{n}_1^T G\boldsymbol{n}_2 \end{bmatrix} = 0. \tag{2.13}$$

Equation (2.13) is a vector equation involving motion parameters R, $\boldsymbol{T}$, S, $\boldsymbol{U}$, and observables $\boldsymbol{n}_0$, $\boldsymbol{n}_1$, and $\boldsymbol{n}_2$. As can be seen, if we scale any of $\boldsymbol{n}_0$, $\boldsymbol{n}_1$, and $\boldsymbol{n}_2$ in (2.13) by a positive number, the equation still holds. Therefore, $\boldsymbol{n}_0$, $\boldsymbol{n}_1$, and $\boldsymbol{n}_2$ can be normalized to be unit vectors. The three scalar equations in (2.13) are linear in the $9 \times 3 = 27$ components of the intermediate parameters (E, F, G). Since $rank([\boldsymbol{n}_0]_\times) = 2$ for $\boldsymbol{n}_0 \neq \boldsymbol{0}$, (2.13) has at most two independent scalar equations. From each line correspondence through three perspective views, we get a set of corresponding projection normals: $\boldsymbol{n}_0$, $\boldsymbol{n}_1$, and $\boldsymbol{n}_2$. If we have at least 13 line correspondences through three views, we might have 26 independent scalar equations. If so, we can solve for the intermediate parameters (E, F, G) up to a scale factor based on (2.13). When a matrix is determined up to a scale factor, we say that it is *essentially* determined. The condition to have 26 independent scalar equations (2.13) is discussed in the next section.

In the subsequent analysis, it is assumed that the intermediate parameters (E, F, G) are essentially determined. For convenience, we solve for the normalized intermediate parameters (E_s, F_s, G_s) with $\|E_s\|^2 + \|F_s\|^2 + \|G_s\|^2 = 1$, such that

$$(E_s, F_s, G_s) = \alpha(E, F, G)$$

where α is an unknown scale factor. The motion parameters are to be determined from the normalized intermediate parameters.

It is easy to see from (2.12) that $\|\boldsymbol{T}\|^2 + \|U\|^2$ is proportional to $\|E\|^2 + \|F\|^2 + \|G\|^2$. If the scene is scaled with respect to the origin by a positive factor of k and the translations $\boldsymbol{T}$ and $\boldsymbol{U}$ are also scaled by k, we get the same images. Therefore, $\|\boldsymbol{T}\|^2 + \|U\|^2$ cannot be determined from the monocular images. For simplicity of notation, we drop the subscript s and let $\|E\|^2 + \|F\|^2 + \|G\|^2 = 1$, with the understanding that (E, F, G) are known only up to a scale factor. As shown later, the rotation matrices are independent of this scale factor.

F. Motion Parameters from Intermediate Parameters

Let $\boldsymbol{V}_i = \boldsymbol{T} \times \boldsymbol{R}_i$, $i = 1, 2, 3$. From (2.12), we have $E^T\boldsymbol{V}_1 = \mathbf{0}$, $F^T\boldsymbol{V}_2 = \mathbf{0}$, and $G^T\boldsymbol{V}_3 = \mathbf{0}$. If the ranks of E, F, G are all equal to two, $\boldsymbol{V}_i$ can be essentially determined from (E, F, G). Then, the translation vector $\boldsymbol{T}$ can be essentially determined by $\boldsymbol{T} \cdot \boldsymbol{V}_i = 0$, $i = 1, 2, 3$. However the ranks of E, F, G are not always equal to 2. The following theorem enumerates all the possible cases.

Theorem 1: Assume $\boldsymbol{T} \neq \mathbf{0}$ and $\boldsymbol{U} \neq \mathbf{0}$. Then, there exist unit vectors $\boldsymbol{V}_1$, $\boldsymbol{V}_2$, and $\boldsymbol{V}_3$ such that

$$E^T\boldsymbol{V}_1 = \mathbf{0} \tag{2.14}$$

$$F^T\boldsymbol{V}_2 = \mathbf{0} \tag{2.15}$$

$$G^T\boldsymbol{V}_3 = \mathbf{0} \tag{2.16}$$

and the ranks of E, F, and G fall into three cases.

Case 1: All of E, F, G have rank two. $\boldsymbol{V}_i$ is then essentially determined. Let

$$A = [\boldsymbol{V}_1 \quad \boldsymbol{V}_2 \quad \boldsymbol{V}_3] \tag{2.17}$$

Then, $rank(A) = 2$, and $\boldsymbol{T}$ is essentially determined by $A^T\boldsymbol{T} = \mathbf{0}$.

Case 2: Two of E, F, G have rank two, and the third has rank one. Without loss of generality, let $rank(E) = 1$. Let $A = [\boldsymbol{V}_2\, \boldsymbol{V}_3]$. If $rank(A) = 2$, $\boldsymbol{T}$ is still essentially determined by $A^T\boldsymbol{T} = \mathbf{0}$. Otherwise, $\boldsymbol{T}$ is essentially determined by $\boldsymbol{T}//(\boldsymbol{E}_i \times \boldsymbol{V}_2) \times \boldsymbol{V}_2$, where $\boldsymbol{E}_i$ is any nonzero column vector of E. $(\boldsymbol{E}_i \times \boldsymbol{V}_2) \times \boldsymbol{V}_2 \neq \mathbf{0}$ is guaranteed.

Case 3: Only one of E, F, G has rank two, and the other two matrices have rank one. Without loss of generality, let $rank(G) = 2$. Then, there are two orthogonal solutions in (2.14) and (2.15), respectively:

$$E^T\boldsymbol{V}_{1a} = \mathbf{0}, \qquad E^T\boldsymbol{V}_{1b} = \mathbf{0},$$
$$F^T\boldsymbol{V}_{2a} = \mathbf{0}, \qquad F^T\boldsymbol{V}_{2b} = \mathbf{0}$$

where $\boldsymbol{V}_{1a} \cdot \boldsymbol{V}_{1b} = 0$, and $\boldsymbol{V}_{2a} \cdot \boldsymbol{V}_{2b} = 0$. One and only one of the two equations

$$\boldsymbol{V}_3 \cdot (\boldsymbol{V}_{1a} \times \boldsymbol{V}_{1b}) = 0 \tag{2.18}$$

and

$$\boldsymbol{V}_3 \cdot (\boldsymbol{V}_{2a} \times \boldsymbol{V}_{2b}) = 0 \tag{2.19}$$

holds. $\boldsymbol{T}//\boldsymbol{V}_{1a} \times \boldsymbol{V}_{1b}$ if (2.18) is true, and $\boldsymbol{T}//\boldsymbol{V}_{2a} \times \boldsymbol{V}_{2b}$ if (2.19) is true.

Proof: See Appendix A.

From Theorem 1, we know that $\boldsymbol{T}$ can be essentially determined. Similarly, if we apply E^T, F^T, G^T to Theorem 1, we know that $\boldsymbol{U}$ can also be essentially determined. In a word, we can determine unit vectors $\mathring{\boldsymbol{T}}_s$ and $\mathring{\boldsymbol{U}}_s$ such that $\mathring{\boldsymbol{T}}_s \times \boldsymbol{T} = \mathbf{0}$, and $\mathring{\boldsymbol{U}}_s \times \boldsymbol{U} = \mathbf{0}$ ("°" at the top of a letter denotes a unit vector.)

The following theorem states the uniqueness of the solution for motion parameters from the intermediate parameters. The condition $\boldsymbol{T} \neq \mathbf{0}$, $\boldsymbol{U} \neq \mathbf{0}$, and $R^T\boldsymbol{T} \neq S^T\boldsymbol{U}$ used in the theorem is called *distinct locations condition.* In Section V, we will see that this condition turns out to be a necessary condition for essentially determining intermediate parameters by (2.13). It is a sufficient condition in the following theorem.

Theorem 2: Given (E, F, G), the solution for R, $\boldsymbol{T}$, S, $\boldsymbol{U}$ is unique, provided $\boldsymbol{T} \neq \mathbf{0}$, $\boldsymbol{U} \neq \mathbf{0}$, and $R^T\boldsymbol{T} \neq S^T\boldsymbol{U}$.

Proof: From Theorem 1, we can determine $\mathring{\boldsymbol{T}}_s$ and $\mathring{\boldsymbol{U}}_s$ such that $\boldsymbol{T} = s_1\|\boldsymbol{T}\|\mathring{\boldsymbol{T}}_s$, and $\boldsymbol{U} = s_2\|\boldsymbol{U}\|\mathring{\boldsymbol{U}}_s$, where $s_1, s_2 \in \{-1, 1\}$. For four combinations of the values of s_1 and s_2, we have four sets of equations:

$$(s_1, s_2): \quad \begin{cases} E = s_1\|\boldsymbol{U}\|\boldsymbol{R}_1\mathring{\boldsymbol{U}}_s^T - s_2\|\boldsymbol{T}\|\mathring{\boldsymbol{T}}_s\,\boldsymbol{S}_1^T \\ F = s_1\|\boldsymbol{U}\|\boldsymbol{R}_2\mathring{\boldsymbol{U}}_s^T - s_2\|\boldsymbol{T}\|\mathring{\boldsymbol{T}}_s\boldsymbol{S}_2^T \\ G = s_1\|\boldsymbol{U}\|\boldsymbol{R}_3\mathring{\boldsymbol{U}}_s^T - s_2\|\boldsymbol{T}\|\mathring{\boldsymbol{T}}_s\boldsymbol{S}_2^T \end{cases} \tag{2.20}$$

Premultiplying both sides of the first equation in (2.20) yields

$$\left[\mathring{\boldsymbol{T}}_s\right]_\times E = s_1\|\boldsymbol{U}\|\left[\mathring{\boldsymbol{T}}_s\right]_\times \boldsymbol{R}_1\mathring{\boldsymbol{U}}_s^T.$$

Postmultiplying both sides by $\mathring{\boldsymbol{U}}_s$ gives

$$\left[\mathring{\boldsymbol{T}}_s\right]_\times E\mathring{\boldsymbol{U}}_s = s_1\|\boldsymbol{U}\|\left[\mathring{\boldsymbol{T}}_s\right]_\times \boldsymbol{R}_1. \tag{2.21}$$

Applying the same operations to the second and the third equations in (2.20) gives the other two equations similar to (2.21). Combining these three equations yields

$$\left[\mathring{\boldsymbol{T}}_s\right]_\times \left[E\mathring{\boldsymbol{U}}_s \quad F\mathring{\boldsymbol{U}}_s \quad G\mathring{\boldsymbol{U}}_s\right] = s_1\|\boldsymbol{U}\|\left[\mathring{\boldsymbol{T}}_s\right]_\times R. \tag{2.22}$$

Since R is a rotation matrix, $\|R\boldsymbol{x}\| = \|\boldsymbol{x}\|$. We get $\|\boldsymbol{U}\|$ from (2.22):

$$\|\boldsymbol{U}\| = \left\|\left[\mathring{\boldsymbol{T}}_s\right]_\times\right\|^{-1} \left\|\left[\mathring{\boldsymbol{T}}_s\right]_\times \left[E\mathring{\boldsymbol{U}}_s \quad F\mathring{\boldsymbol{U}}_s \quad G\mathring{\boldsymbol{U}}_s\right]\right\|. \tag{2.23}$$

Considering the transposed version of E, F, G, similarly, we have

$$\left[\mathring{\boldsymbol{U}}_s\right]_\times \left[E^T\mathring{\boldsymbol{T}}_s \quad F^T\mathring{\boldsymbol{T}}_s \quad G^T\mathring{\boldsymbol{T}}_s\right] = -s_2\|\boldsymbol{T}\|\left[\mathring{\boldsymbol{U}}_s\right]_\times S. \tag{2.24}$$

$\|\boldsymbol{T}\|$ is determined by an equation similar to (2.23). Equations (2.22) and (2.24) both have the form $A = BR$, $rank(B) = 2$ ($[\boldsymbol{x}]_\times$ has a rank 2 if $\boldsymbol{x} \neq \mathbf{0}$, which is shown in Appendix A). Therefore, rotation matrices R and S are uniquely determined in (2.22) and (2.24), respectively. In the presence of noise, we solve for a rotation matrix R in the following

$$\min_{R} \|A - BR\|, \qquad \text{subject to : } R \text{ is a rotation matrix.}$$

The solution of this problem is discussed in [2], [19], [8], [25] and is presented later with the algorithm.

However, there exist four combinations for all the possible signs of (s_1, s_2) in (2.20). The following lemma states that only one combination has a solution for rotation matrices R and S from (2.20).

Lemma 2: Assume $\boldsymbol{T} \neq \boldsymbol{0}$, $\boldsymbol{U} \neq \boldsymbol{0}$, and $R^T\boldsymbol{T} \neq S^T\boldsymbol{U}$. Only one assignment for (s_1, s_2), $s_1, s_2 \in \{-1, 1\}$ has a solution for rotation matrices R and S from (2.20).

Proof: See Appendix C.

By substituting into (2.20) the four assignments for (s_1, s_2), we get a unique solution R and S and the assignment of (s_1, s_2), which is the one that satisfies (2.20). □

On the other hand, (E, F, G) can only be essentially determined, i.e., up to a scale factor. From (2.22)–(2.24), it is easy to see that the scale factor does not affect the solution of the rotation matrices R and S. However, the translation vector pair $(\boldsymbol{T}, \boldsymbol{U})$ is essentially determined, which implies that the ratio between $\|\boldsymbol{T}\|$ and $\|\boldsymbol{U}\|$ is determined. We can choose any sign for (E, F, G) and solve for the translation vector pair to get $\boldsymbol{T}_s$ and $\boldsymbol{U}_s$ such that $(\boldsymbol{T}, \boldsymbol{U}) = \alpha(\boldsymbol{T}_s, \boldsymbol{U}_s)$ with unknown α. The absolute value of α cannot be determined from monocular images. The sign of α will be determined next.

G. Structure and Sign of Translation Vectors

From (2.4)–(2.6), we get $\boldsymbol{l} \cdot \boldsymbol{n}_0 = 0$, $\boldsymbol{l} \cdot R^{-1}\boldsymbol{n}_1 = 0$, and $\boldsymbol{l} \cdot S^{-1}\boldsymbol{n}_2 = 0$. For each line, we solve for a unit vector $\hat{\boldsymbol{l}}$ such that $\boldsymbol{l} // \hat{\boldsymbol{l}}$ in the following:

$$\min_{\hat{\boldsymbol{l}}} \left\| \left[\boldsymbol{n}_0 \quad R^{-1}\boldsymbol{n}_1 \quad S^{-1}\boldsymbol{n}_2\right]^T \hat{\boldsymbol{l}} \right\|, \qquad \text{subject to : } \left\|\hat{\boldsymbol{l}}\right\| = 1. \tag{2.25}$$

If the rank of $\left[\boldsymbol{n}_0 \; R^{-1}\boldsymbol{n}_1 \; S^{-1}\boldsymbol{n}_2\right]$ is no more than one, the line position cannot be recovered.

For each line, let $\boldsymbol{x}_p$ be a point on the line that is the closest to the origin. $d_0 \triangleq \|\boldsymbol{x}_p\|$ is the positive distance of the line to the origin. Since $\boldsymbol{x}_p \cdot \boldsymbol{l} = 0$, from (2.4), we have

$$\|\boldsymbol{n}_0\| = \|\boldsymbol{x}_p \times \boldsymbol{l}\| = \|\boldsymbol{x}_p\| \, \|\boldsymbol{l}\|. \tag{2.26}$$

Although we use (2.4)–(2.6) to define the projection normals, the scale factor of those normals is immaterial since it will be canceled out later in (2.27). Using (2.10) and (2.26) yields

$$\begin{aligned} |\boldsymbol{T} \cdot \boldsymbol{n}_1| &= \|\boldsymbol{l}\|^{-1} \left\|\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1\right\| \\ &= \|\boldsymbol{x}_p\| \, \|\boldsymbol{n}_0\|^{-1} \left\|\boldsymbol{n}_0 \times R^{-1}\boldsymbol{n}_1\right\|. \end{aligned}$$

Dividing both sides by $\|\boldsymbol{n}_1\|$ gives the distance of the line to the origin

$$d_0 = \|\boldsymbol{x}_p\| = \left\|\mathring{\boldsymbol{n}}_0 \times R^{-1}\mathring{\boldsymbol{n}}_1\right\|^{-1} \left|\boldsymbol{T} \cdot \mathring{\boldsymbol{n}}_1\right|. \tag{2.27}$$

Notice that d_0 is proportional to $\|\boldsymbol{T}\|$. When $\boldsymbol{T}_s$ replaces for $\boldsymbol{T}$ in (2.27), $\|\boldsymbol{x}_p\|$ is essentially determined.

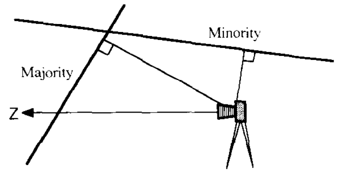


Fig. 4. Majority positive depths assumption. Majority of all the visible lines have their closest points in front of the camera. Only a minority of the visible lines have their closest points in back of the camera.

Let $\mathring{\boldsymbol{v}}$ be a unit vector that is parallel to $\boldsymbol{x}_p$ and always points to the positive $\boldsymbol{Z}$ direction, that is, $\mathring{\boldsymbol{v}} = \pm\|\boldsymbol{n}_0 \times \boldsymbol{l}\|^{-1}\boldsymbol{n}_0 \times \boldsymbol{l}$, such that $\boldsymbol{Z} \cdot \mathring{\boldsymbol{v}} > 0$. Then, $\boldsymbol{x}_p = \pm d_0\mathring{\boldsymbol{v}}$.

To determine the sign for the translation vectors, first consider the motion equation (2.1).

$$\boldsymbol{x}_1 = R\boldsymbol{x}_0 + \boldsymbol{T}.$$

Multiplying both sides by -1, we get

$$-\boldsymbol{x}_1 = R(-\boldsymbol{x}_0) + (-\boldsymbol{T}). \tag{2.28}$$

Equation (2.28) implies that when a point $\boldsymbol{x}_0$ is rotated by R and translated by $\boldsymbol{T}$ to a point $\boldsymbol{x}_1$, its minor image (with respect to the origin) $-\boldsymbol{x}_0$ is rotated by R and translated by $-\boldsymbol{T}$. The original line and the mirror image line produce the same images through the projection plane. Therefore, if $\boldsymbol{T}$ and $\boldsymbol{U}$ are the true translation vectors, $-\boldsymbol{T}$ and $-\boldsymbol{U}$ are the translation vectors of the mirror image. Obviously, if a point is located in front of the camera, its mirror image is at the back of the camera. Therefore, if we assume that all of the visible scene is in front of the camera, then the translation that corresponds to the structure located at back of the camera is not the correct one. As to the lines, what is the criterion for a line to be in front of the camera? We can assume that most lines in front of the camera have $\boldsymbol{x}_p$, which is the closest to the origin on the line, with a positive (depth) z component. This assumption is called *majority positive depths assumption* (see Fig. 4). It is usually satisfied because most lines whose closest point $\boldsymbol{x}_p$ has a negative depth are not visible through the camera lens. A few exceptional visible lines may have negative depth at $\boldsymbol{x}_p$. For example, when we look horizontally in a car that is running downhill, the two curb lines are visible, but $\boldsymbol{x}_p$ of each of these two lines has negative z components since $\boldsymbol{x}_p$ is at the back of the eyes. However, those lines with negative z components usually constitute a minority among all visible lines.

Let $\boldsymbol{T}_s = s\boldsymbol{T}$, where $s \in \{1, -1\}$ is unknown. Assume a line at time t_0 with $\boldsymbol{x}_p$, which is the point closest to the origin. At time $\boldsymbol{t}_1$, its new position vector should be orthogonal to the new projection normal

$$\boldsymbol{n}_1 \cdot (R\boldsymbol{x}_p + \boldsymbol{T}) = \boldsymbol{n}_1 \cdot (R\boldsymbol{x}_p + s\boldsymbol{T}_s) = 0. \tag{2.29}$$

Assume that $\boldsymbol{x}_p$ has a positive z component at time $\boldsymbol{t}_0$, $\boldsymbol{x}_p = \|\boldsymbol{x}_p\|\mathring{\boldsymbol{v}}$. From (2.29), we have

$$\boldsymbol{n}_1 \cdot \left(\|\boldsymbol{x}_p\|R\mathring{\boldsymbol{v}} + s\boldsymbol{T}_s\right) = 0. \tag{2.30}$$

Note that (2.30) cannot hold for both $s = 1$ and $s = -1$ unless $\boldsymbol{n}_1 \cdot \boldsymbol{T}_s = 0$, which is rare. In a word, if $\boldsymbol{x}_p$ has a positive z component, the correct sign for s satisfies (2.30), and the incorrect one generally does not. If $\boldsymbol{x}_p$ has a negative z component, we have $\boldsymbol{x}_p = -\|\boldsymbol{x}_p\|\mathring{\boldsymbol{v}}$, which yields

$$\boldsymbol{n}_1 \cdot \left(-\|\boldsymbol{x}_p\|R\mathring{\boldsymbol{v}} + s\boldsymbol{T}\right) = 0. \tag{2.31}$$

In other words, a point with a negative z component satisfies (2.30) if the sign for s is reversed.

We determine the sign s in the following way. For each line, if $s = 1$ satisfies (2.30), a vote is added to the set POS, and if $s = -1$ satisfies (2.30), a vote is added to the set NEG. After voting by all lines, if POS has more votes than NEG, $s = 1$ supports the majority positive depths assumption. Therefore, we let $s = 1$ and $\boldsymbol{T} = +\boldsymbol{T}_s$. All the lines that voted for POS should have $\boldsymbol{x}_p = +\|\boldsymbol{x}_p\|\mathring{\boldsymbol{v}}$ since they are majority, and the results are consistent with (2.30). For all the lines that voted for NEG should have $\boldsymbol{x}_p = -\|\boldsymbol{x}_p\|\mathring{\boldsymbol{v}}$ since they are minority, and the results are consistent with (2.31).

Otherwise, POS has less votes. This means that $s = -1$ supports the majority positive depths assumption. Therefore $s = -1$, and $\boldsymbol{T} = -\boldsymbol{T}_s$. All the lines that voted for POS should have $\boldsymbol{x}_p = -\|\boldsymbol{x}_p\|\mathring{\boldsymbol{v}}$ since they are minority, and the results are consistent with (2.31). All the lines that voted for NEG should have $\boldsymbol{x}_p = \|\boldsymbol{x}_p\|\mathring{\boldsymbol{v}}$ since they are majority, and again, the results are consistent with (2.30). Thus, the sign of $\boldsymbol{T}$ and the locations of the lines can be determined based on the majority positive depths assumption.

The motion from t_0 to t_2 can be analyzed in a similar way.

H. In the Presence of Noise

Since short lines in the images are generally not as reliably determined as long lines, less weight should be assigned to the short lines in (2.13) when solving for E, F, G. Let the length of the lines in the images at time t_i be l_i, $i = 0, 1, 2$. A simple weight for the line can be

$$\left(l_0^{-1} + l_1^{-1} + l_2^{-1}\right)^{-1}. \tag{2.32}$$

Since only two of the three scalar equations in (2.13) are independent, any row of $[\boldsymbol{n}_0]_\times$ could be zero vector, which results in a trivial $0 = 0$ equations. To ensure that none of the two independent equations is missed, we can use all three equations of (2.13). (This is useful in the presence of noise.) These three equations are scaled by the weight in (2.32) in the system of linear equations formed by combining (2.13) for all the lines.

Since a noise-corrupted matrix almost has a full rank, the conditions on the rank of the matrices should then be modified accordingly. A discussion of the sensitivity of the eigenvectors to the perturbation of the matrix can be found in [25]. A rough measurement for the error of the eigenvector associated with the smallest eigenvalue λ_1 is $(\lambda_1 - \lambda_2)^{-1}$, where λ_2 is the second smallest eigenvalue. The solution of $\boldsymbol{V}_i$ $i = 1, 2, 3$ in (2.14)–(2.16) is the eigenvector of EE^T, FF^T, GG^T, respectively, associated with the smallest eigenvalue. The reliability of those solutions is roughly proportional to the difference between the smallest and the second smallest eigenvalues. Case 1 and Case 2 in Theorem 1 can be combined by using a weighted A in (2.17). Let the three column vectors of A be weighted by the difference of the two smallest eigenvalues of EE^T, FF^T, and GG^T, respectively. For example, if $rank(E)$ is close to one, the corresponding weight in A is close to zero, which is Case 2.

In determining the distance d_0, the motion from t_0 to t_1 and that from t_0 to t_2 can both be used to enhance the robustness.

Obviously, the simple weighting schemes discussed in this subsection are *ad hoc*. More complete weighting methods will be discussed in Section V. The objective here is to use simple weighting while still keeping the algorithm linear. In our simulation, we observed considerable improvements by using the simple weighting schemes discussed above.

I. Algorithm

Now, we are ready to present the algorithm. In the algorithm, ε denotes a small positive threshold to accommodate noise. Without noise, ε should be zero. With noise, ε can be estimated by the error estimation approach in [25] or determined empirically. Though ε should be different in different parts of the algorithm, a single ε will be used for the simplicity of notation.

1) Solving for (E, F, G) Up to a Scale Factor: Given n line correspondences through three views. Let the unit projection normal at time t_i be $\boldsymbol{n}_i$, $i = 0, 1, 2$. Solve for (E, F, G) in the following

$$\min_{(E,F,G)} \sum_{\text{lines}} \text{weight} \left\| [\boldsymbol{n}_0]_\times \begin{bmatrix} \boldsymbol{n}_1^T E \boldsymbol{n}_2 \\ \boldsymbol{n}_1^T F \boldsymbol{n}_2 \\ \boldsymbol{n}_1^T G \boldsymbol{n}_2 \end{bmatrix} \right\|^2 \tag{2.33}$$

subject to $\|E\|^2 + \|F\|^2 + \|G\|^2 = 1$, where the weight for each line is given in (2.32). (2.33) can be written in the form

$$\min_{y} \|D\boldsymbol{y}\| \qquad \text{subject to: } \|\boldsymbol{y}\| = 1 \tag{2.34}$$

where D is a $3n$ by 27 matrix determined from projection normals, and $\boldsymbol{y}$ is a 27-dimensional unit vector. The solution for unit vector $\boldsymbol{y}$ is the unit eigenvector of $D^T D$ associated with the smallest eigenvalue.

2) Determining unit vectors $\mathring{\boldsymbol{T}}_s$ and $\mathring{\boldsymbol{U}}_s$ such that $\boldsymbol{T}//\mathring{\boldsymbol{T}}_s$ and $\boldsymbol{U}//\mathring{\boldsymbol{U}}_s$: Let $H_e \triangleq [\boldsymbol{h}_{e1}\ \boldsymbol{h}_{e2}\ \boldsymbol{h}_{e3}]$, $F_f \triangleq [\boldsymbol{h}_{f1}\ \boldsymbol{h}_{f2}\ \boldsymbol{h}_{f3}]$, and $H_g \triangleq [\boldsymbol{h}_{g1}\ \boldsymbol{h}_{g2}\ \boldsymbol{h}_{g3}]$ be orthogonal matrices such that

$$\begin{aligned} H_e^T EE^T H_e &= \text{diag}(\lambda_{e1}, \lambda_{e2}, \lambda_{e3}), & \lambda_{e1} \le \lambda_{e2} \le \lambda_{e3} \\ H_f^T FF^T H_f &= \text{diag}(\lambda_{f1}, \lambda_{f2}, \lambda_{f3}), & \lambda_{f1} \le \lambda_{f2} \le \lambda_{f3} \\ H_g^T GG^T H_g &= \text{diag}(\lambda_{g1}, \lambda_{g2}, \lambda_{g3}), & \lambda_{g1} \le \lambda_{g2} \le \lambda_{g3}. \end{aligned}$$

Case 1: The median of the set $C = \{\lambda_{e2}, \lambda_{f2}, \lambda_{g2}\}$ is larger than ε.

Let $A = [(\lambda_{e2} - \lambda_{e1})\boldsymbol{h}_{e1}\ (\lambda_{f2} - \lambda_{f1})\boldsymbol{h}_{f1}\ (\lambda_{g2} - \lambda_{g1})\boldsymbol{h}_{g1}]$.

a) If the second smallest eigenvalue of $A^T A$ is larger than $\varepsilon(rank(A) \ge 2)$, $\mathring{\boldsymbol{T}}_s$ is determined up to a scale factor by

$$\min_{\mathring{\boldsymbol{T}}_s} \left\|A^T \mathring{\boldsymbol{T}}_s\right\|, \qquad \text{subject to: } \left\|\mathring{\boldsymbol{T}}_s\right\| = 1. \tag{2.35}$$

b) Otherwise $(rank(A) \approx 1)$, determine the smallest number in set C. If λ_{e2} is the smallest in set C, then

$\mathring{\boldsymbol{T}}_s // (\boldsymbol{E}_i \times \boldsymbol{h}_{f1}) \times \boldsymbol{h}_{f1}$, where $\boldsymbol{E}_i$ is a nonzero column vector of E. If λ_{f2} or λ_{g2} is the smallest in $C, \mathring{\boldsymbol{T}}_s$ is determined by a similar equation (circularly rotating e, f, g and E, F, G).

Case 2: The median of the set $C = \{\lambda_{e2}, \lambda_{f2}, \lambda_{g2}\}$ is not larger than ε. Determine the maximum of the set C. Without loss of generality, assume λ_{g2} is the maximum:

$$\mathring{\boldsymbol{T}}_s = \begin{cases} \boldsymbol{h}_{e1} \times \boldsymbol{h}_{e2} \\ \qquad \text{if } |\boldsymbol{h}_{g1} \cdot (\boldsymbol{h}_{e1} \times \boldsymbol{h}_{e2})| < |\boldsymbol{h}_{g1} \cdot (\boldsymbol{h}_{f1} \times \boldsymbol{h}_{f2})| \\ \boldsymbol{h}_{f1} \times \boldsymbol{h}_{f2} \quad \text{otherwise} \end{cases}$$

Replacing E, F, G by E^T, F^T, G^T, similarly determine $\mathring{\boldsymbol{U}}_s$.

3) Determining R and S: Let

$$G_R = \left[\mathring{\boldsymbol{T}}_s\right]_\times \left[E\mathring{\boldsymbol{U}}_s \quad F\mathring{\boldsymbol{U}}_s \quad G\mathring{\boldsymbol{U}}_s\right]$$
$$G_S = \left[\mathring{\boldsymbol{U}}_s\right]_\times \left[E^T\mathring{\boldsymbol{T}}_s \quad F^T\mathring{\boldsymbol{T}}_s \quad G^T\mathring{\boldsymbol{T}}_s\right]$$

and $\|\boldsymbol{U}\| = \|G_R\|/\sqrt{2}$, $\|\boldsymbol{T}\| = \|G_S\|/\sqrt{2}$. Then, let $G_R \leftarrow \|\boldsymbol{U}\|^{-1} G_R$, and $G_S \leftarrow \|\boldsymbol{T}\|^{-1} G_S$. Solve for R_p, R_n, S_p, and S_n in the following:

$$\min_{R_p} \left\| G_R - \left[\mathring{\boldsymbol{T}}_s\right]_\times R_p \right\| \qquad \min_{R_n} \left\| -G_R - \left[\mathring{\boldsymbol{T}}_s\right]_\times R_n \right\| \tag{2.36}$$

$$\min_{S_p} \left\| G_S - \left[\mathring{\boldsymbol{U}}_s\right]_\times S_p \right\| \qquad \min_{S_n} \left\| -G_S - \left[\mathring{\boldsymbol{U}}_s\right]_\times S_n \right\| \tag{2.37}$$

subject to that R_p, R_n, S_p, and S_n are rotation matrices. Both (2.36) and (2.37) have the form (noticing $\|D - CR\| = \|R^T C^T - D^T\|$):

$$\min_R \|RC - D\|, \qquad \text{subject to : } R \text{ is a rotation matrix} \tag{2.38}$$

where $C = [\boldsymbol{C}_1 \ \boldsymbol{C}_2 \ \boldsymbol{C}_3]$, $D = [\boldsymbol{D}_1 \ \boldsymbol{D}_2 \ \boldsymbol{D}_3]$. The solution of (2.38) is as follows:

Define a $4x4$ matrix B by

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

where

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

Let $\boldsymbol{q} = (q_0, q_1, q_2, q_3)^T$ be the unit eigenvector of B associated with the smallest eigenvalue. The solution of rotation matrix R in (2.38) is found at the bottom of this page.

Substituting four assignments

$$(1, 1, R_p, S_p), \qquad (-1, 1, R_n, S_p)$$
$$(1, 1, R_p, S_n), \qquad (-1, -1, R_n, S_n)$$

for (s_t, s_u, R, S) in

$$\begin{cases} E = s_t\|\boldsymbol{U}\|\boldsymbol{R}_1\mathring{\boldsymbol{U}}_s^T - s_u\|\boldsymbol{T}\|\mathring{\boldsymbol{T}}_s\boldsymbol{S}_1^T \\ F = s_t\|\boldsymbol{U}\|\boldsymbol{R}_2\mathring{\boldsymbol{U}}_s^T - s_u\|\boldsymbol{T}\|\mathring{\boldsymbol{T}}_s\boldsymbol{S}_2^T \\ G = s_t\|\boldsymbol{U}\|\boldsymbol{R}_2\mathring{\boldsymbol{U}}_s^T - s_u\|\boldsymbol{T}\|\mathring{\boldsymbol{T}}_s\boldsymbol{S}_2^T \end{cases} \tag{2.39}$$

the assignment best satisfying (2.39), in the Euclidean norm sense, gives the correct assignment for (s_t, s_u, R, S). Then

$$\boldsymbol{T} = s_t\|\boldsymbol{T}\|\mathring{\boldsymbol{T}}_s, \qquad \boldsymbol{U} = s_u\|\boldsymbol{U}\|\mathring{\boldsymbol{U}}_s.$$

4) Determining Structure $\mathring{\boldsymbol{l}}$ *and* $\boldsymbol{x}_p$: For each line, solve for the direction of the line represented by a vector $\mathring{\boldsymbol{l}}$ in the following:

$$\min_{\mathring{\boldsymbol{l}}} \left\| [\boldsymbol{n}_0 \quad R^{-1}\boldsymbol{n}_1 \quad S^{-1}\boldsymbol{n}_2]\mathring{\boldsymbol{l}} \right\|$$
$$\text{subject to : } \left\|\mathring{\boldsymbol{l}}\right\| = 1.$$

Then

$$d_0 = \frac{\left|\boldsymbol{T} \cdot \mathring{\boldsymbol{n}}_1\right|}{2\left\|\mathring{\boldsymbol{n}}_0 \times R^{-1}\mathring{\boldsymbol{n}}_1\right\|} + \frac{\left|\boldsymbol{U} \cdot \mathring{\boldsymbol{n}}_2\right|}{2\left\|\mathring{\boldsymbol{n}}_0 \times S^{-1}\mathring{\boldsymbol{n}}_2\right\|}$$

and

$$\mathring{\boldsymbol{v}} = \pm \frac{\mathring{\boldsymbol{n}}_0 \times \mathring{\boldsymbol{l}}}{\left\|\mathring{\boldsymbol{n}}_0 \times \mathring{\boldsymbol{l}}\right\|}$$

where the sign is such that the third component of $\mathring{\boldsymbol{v}}$ is nonnegative.

Let POS and NEG be empty sets. For each line i, do the following: If

$$\left|\boldsymbol{n}_1 \cdot \left(d_0 R\mathring{\boldsymbol{v}} + \boldsymbol{T}\right)\right| + \left|\boldsymbol{n}_2 \cdot \left(d_0 S\mathring{\boldsymbol{v}} + \boldsymbol{U}\right)\right|$$
$$< \left|\boldsymbol{n}_1 \cdot \left(d_0 R\mathring{\boldsymbol{v}} - \boldsymbol{T}\right)\right| + \left|\boldsymbol{n}_2 \cdot \left(d_0 S\mathring{\boldsymbol{v}} - \boldsymbol{U}\right)\right|$$

i is added into the set POS. Otherwise, i is added to the set NEG.

Finally, if $\|POS\| > \|NEG\|$, for each line i

$$\boldsymbol{x}_p = \begin{cases} d_0\mathring{\boldsymbol{v}} & \text{if } i \in POS \\ -d_0\mathring{\boldsymbol{v}} & \text{otherwise.} \end{cases}$$

Otherwise, if $\|POS\| < \|NEG\|$

$$\boldsymbol{T} \leftarrow -\boldsymbol{T}, \qquad \boldsymbol{U} \leftarrow -\boldsymbol{U}.$$

For each line i

$$\boldsymbol{x}_p = \begin{cases} -d_0\mathring{\boldsymbol{v}} & \text{if } i \in POS \\ d_0\mathring{\boldsymbol{v}} & \text{otherwise.} \end{cases}$$

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

III. Degeneracy

In the last section, it is established by Theorem 2 that as long as $\boldsymbol{T} \neq \mathbf{0}$, $\boldsymbol{U} \neq \mathbf{0}$ and $\boldsymbol{T}^T R \neq \boldsymbol{U}^T S$, the solution of motion parameters from the intermediate parameters (E, F, G) is unique. The fact that (E, F, G) can only be determined up to a scale factor does not affect the solution of the rotation matrices R and S. The direction of the translation and the structure of the lines can be determined based on the majority positive depths assumption. Therefore, the translation and the closest points on the lines are determined up to a positive scale factor.

First, let us see what the condition

$$\boldsymbol{T} \neq \mathbf{0}, \qquad \boldsymbol{U} \neq \mathbf{0}, \qquad \boldsymbol{T}^T R \neq \boldsymbol{U}^T S \tag{3.1}$$

means. For a more intuitive interpretation, we consider the case where the scene is stationary and the camera is moving. $\boldsymbol{x}_1$ in (2.1) is the position of the point at time t_1 in a coordinate system fixed on the camera. If the scene is stationary, the point $\boldsymbol{x}_0$ is fixed, and the transformation that transforms $\boldsymbol{x}_1$ to $\boldsymbol{x}_0$ corresponds to the motion of the camera. From (2.1) we have

$$\boldsymbol{x}_0 = R^{-1}\boldsymbol{x}_1 - R^T\boldsymbol{T}. \tag{3.2}$$

Therefore, the motion of the camera is a rotation R^{-1} followed by a translation $-R^{-1}\boldsymbol{T}$. Since the projection center of the camera at time t_0 is at the origin and the rotation is about the origin, the position of the projection center at time t_1 is at $\boldsymbol{O}_1 = -R^T\boldsymbol{T}$. Similarly the position of the projection center at time t_2 is at $\boldsymbol{O}_2 = -S^T\boldsymbol{U}$. Thus, the condition in (3.1) is equivalent to the condition

$$\boldsymbol{O}_1 \neq \mathbf{0}, \qquad \boldsymbol{O}_2 \neq \mathbf{0}, \qquad \boldsymbol{O}_1 \neq \boldsymbol{O}_2. \tag{3.3}$$

That is to say that any two positions of the projection center of the camera do not coincide, or in other words, the translation between any two views does not vanish. Therefore, the condition in (3.1) or (3.3) is called *distinct locations condition.*

The intermediate parameters (E, F, G) are essentially determined by (2.34) or equivalently by (2.13) if and only if the rank of D is not under 26. If (E, F, G) are not essentially determined by (2.13), we say that degeneracy occurs. The degenerarcy condition can be tested by calculating the rank of D. However, in the presence of noise, the rank of D is mostly full. The method of error estimation in [25] can be used to access the accuracy of the intermediate parameters and the final motion parameters. If the estimated error in the solution is large, we say that degeneracy or near degeneracy occurs.

The following theorem gives the necessary and sufficient conditions for degeneracy in terms of 3-D line configurations at time t_0 and the motion parameters.

Theorem 3: (E, F, G) is not essentially determined by (2.13) or equivalently, $rank(D) < 26$ in (2.34) if and only if there exist no trivial parameters $(\tilde{E}, \tilde{F}, \tilde{G})$ such that

$$[\boldsymbol{n}]_\times \begin{bmatrix} ((\boldsymbol{x}_p - \boldsymbol{O}_1) \times \boldsymbol{l})^T \tilde{E}((\boldsymbol{x}_p - \tilde{\boldsymbol{O}}_2) \times \boldsymbol{l}) \\ ((\boldsymbol{x}_p - \boldsymbol{O}_1) \times \boldsymbol{l})^T \tilde{F}((\boldsymbol{x}_p - \tilde{\boldsymbol{O}}_2) \times \boldsymbol{l}) \\ ((\boldsymbol{x}_p - \boldsymbol{O}_1) \times \boldsymbol{l})^T \tilde{G}((\boldsymbol{x}_p - \tilde{\boldsymbol{O}}_2) \times \boldsymbol{l}) \end{bmatrix} = \mathbf{0} \tag{3.4}$$

is satisfied for all lines $\boldsymbol{x} = \boldsymbol{x}_p + k\boldsymbol{l}$ at time t_0. $(\tilde{E}, \tilde{F}, \tilde{G})$ is trivial if and only if

$$\left(\tilde{E}, \tilde{F}, \tilde{G}\right) = \alpha\left(P_1 - Q_1^T, P_2 - Q_2^T, P_3 - Q_3^T\right) \tag{3.5}$$

for some real number α, where P_i and Q_i are matrices with the ith column being $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$, respectively, and the other columns are zero vectors.

Proof: See Appendix E.

It can be seen that the degeneracy depends on two factors: one is the motion parameters, and the other is the configuration of 3-D lines. We have the following corollary:

Corollary 1: If the distinct locations condition is not satisfied, the intermediate parameters are not essentially determined by (2.13).

Proof: See Appendix F.

Therefore, if the distinct locations condition is not satisfied (E, F, G) cannot be essentially determined by (2.13) regardless of the structure of the lines. If the distinct locations condition is satisfied and (2.13) is still degenerate, we say the line configuration is degenerate. The following corollary gives an example of degenerate line configurations.

Corollary 2: If the directions of lines are coplanar, the line configuration is degenerate.

Proof: Let all the lines be orthogonal to a vector $\boldsymbol{v}$. First, we want to prove $[\boldsymbol{l}]_\times[\boldsymbol{v}]_\times[\boldsymbol{l}]_\times = \mathbf{0}$. In fact, it is easy to verify the identity

$$[\boldsymbol{x}]_\times[\boldsymbol{y}]_\times = (\boldsymbol{x} \cdot \boldsymbol{y})\boldsymbol{I} - \boldsymbol{y}\boldsymbol{x}^T.$$

Then, it follows that

$$[\boldsymbol{l}]_\times[\boldsymbol{v}]_\times[\boldsymbol{l}]_\times = 0l - \boldsymbol{v}\boldsymbol{l}^T[\boldsymbol{l}]_\times = -\boldsymbol{v}\mathbf{0}^T = \mathbf{0}.$$

Then, letting $\tilde{E} = \tilde{F} = \tilde{G} = [\boldsymbol{v}]_\times$, the second column vector on the left-hand side of (3.4) vanishes. In fact

$$((\boldsymbol{x}_p - \boldsymbol{O}_1) \times \boldsymbol{l})^T[\boldsymbol{v}]_\times((\boldsymbol{x}_p - \boldsymbol{O}_2) \times \boldsymbol{l}) \\ = (\boldsymbol{x}_p - \boldsymbol{O}_1)^T[\boldsymbol{l}]_\times[\boldsymbol{v}]_\times[\boldsymbol{l}]_\times(\boldsymbol{O}_2 - \boldsymbol{x}_p) = \mathbf{0}.$$

In other words, (3.4) is satisfied for nontrivial $(\tilde{E}, \tilde{F}, \tilde{G})$. Therefore, the configuration is degenerate. □

If all the lines lie in a plane, the direction of liens must be coplanar (they are orthogonal to the normal of the plane). Therefore, a planar scene is a degenerate case. Obviously, the case of coplanar directions is more general since lines do not have to be in a plane.

For lines whose matrix $\left[\boldsymbol{n}_0 \; R^{-1}\boldsymbol{n}_1 \; S^{-1}\boldsymbol{n}_2\right]$ in (2.25) has rank no more than one, the line position cannot be recovered. We neglect this line. This happens if and only if the line lies in the plane determined by the origin and $\boldsymbol{O}_1$ and $\boldsymbol{O}_2$.

From Theorems 2 and 3 and the majority positive depths assumption, we come to the conclusion that if the distinct locations condition is satisfied and the line structure is not degenerate, the motion parameters and the structure of the line can be uniquely determined (up to a scale factor for translation and line distances).

Another form of necessary and sufficient condition is presented in the following theorem.

Theorem 4: Assume 13 line correspondences at time t_0, with the ith line being represented by $\boldsymbol{x} = \boldsymbol{x}_{pi} + \boldsymbol{l}_i$ ($\boldsymbol{x}_{pi}$ is the point closest to the origin) and the normal to the projection plane that passes through the line and the projection center $\boldsymbol{O}_k$ being $\tilde{\boldsymbol{n}}_{ki}$, $k = 1, 2$, $i = 1, 2 \ldots 13$. Then, (E, F, G) is not essentially determined in (2.13), or equivalently, $rank(D) < 26$ in (2.34), if and only if there exist no a_i, b_i, $i = 1, 2, 3, \ldots, 13$, not all of which are zeroes, such that

$$\sum_{i=1}^{13} (a_i \boldsymbol{x}_{pi} + b_i \boldsymbol{l}_i) \tilde{\boldsymbol{n}}_{2i} \tilde{\boldsymbol{n}}_{3i} = 0 \tag{3.6}$$

where the concatenation of two vectors, like $\boldsymbol{pq}$, denotes a tensor out product. For m-dimensional vector $\boldsymbol{p}$ and n-dimensional vector $\boldsymbol{q}$, $\boldsymbol{pq}$ has mn components, which are products of all the possible combinations between one element in $\boldsymbol{p}$ and one in $\boldsymbol{q}$.

Proof: See Appendix G.

Before we end this section, two points should be emphasized here: 1) The fact that the linear algorithm cannot give a unique solution under degenerate line configurations dose not mean that no algorithms can solve the problem under these configurations. Other algorithms, e.g., some nonlinear algorithms, might still be able to reach a correct solution under these configurations. However, as we mentioned, the uniqueness of solution by a nonlinear algorithm is unknown, and the correct convergence of a nonlinear algorithm is not guaranteed; 2) we do not assume any *a priori* knowledge of the scene other than rigidity. However, if we know some additional information, some of the degenerate configurations can be solved. For example, if we know that the scene is planar, the intersection of those coplanar lines gives feature points (no matter whether the lines meet at those points or not), and the planar point-based algorithm can be used to solve the problem.

IV. Optimization

The closed-form solution discussed above makes use of the least squares criterion in several steps. However, the solution is not overall optimum. The first reason is that in solving (2.33), we have neglected the dependency among the components of the intermediate parameters (E, F, G). As defined in (2.12), (E, F, G) has only 12 degrees of freedom (six for each motion), but a free (E, F, G) has 27 degrees of freedom. Some of the dependency has been taken into account in the later steps of the linear algorithm, as in (2.35)–(2.37), but this later recovery is not as effective as imposing the dependency in the first place. The second reason is that for the linear algorithm, we do not consider noise distributions or different amounts of noise in different components of the data. This results in a suboptimal solution in the presence of noise, although the solution is analytical. Our objective in this section is to obtain "the best possible" estimates from the noise corrupted data.

A. Lines from Pixels

An image line consists of edge points. The image position of the line is typically determined from those edge points by a line fitting algorithm. Since lines are used as primitives, it is natural and efficient to consider the parameters of lines as the variables of observations. The representation of lines should be such that the error arising from the line finding process can be naturally modeled. In most line finding processes, the sequence of edge points is indexed along one of the two pixel coordinate directions. Which direction is used for indexing depends on which dimension has a longer projection from the line segment. Mathematically, such a line in the (u, v) image plane is commonly represented by

$$v = au + b \tag{4.1}$$

or

$$u = av + b \tag{4.2}$$

where a and b are parameters of the line. For each line, the selection of (4.1) or (4.2) is such that $|a| \le 1$ (a 1-b flag for each line indicates which is used). The variable u in (4.1) and v in (4.2) can be considered as the image coordinates of edge pixels (see Fig. 5). While the variable on the right-hand side runs from the first pixel location to the last, all the line pixels are enumerated. In any case, using least squares fitting to these edge pixels, a line parameter vector $\boldsymbol{a} \triangleq (a, b)^T$ is "observed." Certainly, it is not appropriate to assume that the parameters here are equally reliable. We need to evaluate the covariance matrix of the error in the line parameter vector. Suppose that the edge pixel (u_i, v_i) in a line is corrupted by noise in v direction in the case of (4.1):

$$v_i = au_i + b + \delta_{v_i} \tag{4.3}$$

or in u direction in the case of (4.2):

$$u_i = av_i + b + \delta_{u_i}. \tag{4.4}$$

This agrees with common methods to determine a line: An edge point on a line is indexed along a pixel row or column (see Fig. 5). If a least squares fitting is employed to edge points, for (4.3) we have

$$A\boldsymbol{a} = \boldsymbol{V} + \boldsymbol{\delta} \tag{4.5}$$

where

$$A = \begin{bmatrix} u_1 & u_2 & \cdots & u_n \\ 1 & 1 & \cdots & 1 \end{bmatrix}^T. \tag{4.6}$$

$\boldsymbol{V} = (v_1, v_2, \cdots, v_n)$, and δ is the corresponding noise vector. Similar results can be obtained for (4.4).

To give an optimal solution we use Gauss-Markov theorem: Suppose

$$\boldsymbol{y} = A\boldsymbol{m} + \boldsymbol{\delta}_y \tag{4.7}$$

where $\boldsymbol{\delta}_y$ is a random vector with zero mean, $\mathbb{E}\boldsymbol{\delta}_y = 0$ and covariance matrix

$$\Gamma_y = \mathbb{E}\boldsymbol{\delta}_y\boldsymbol{\delta}_y^T. \tag{4.8}$$

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

$$\hat{\boldsymbol{m}} = \left(A^T\Gamma_y^{-1}A\right)^{-1}A^T\Gamma_y^{-1}\boldsymbol{y} \tag{4.9}$$

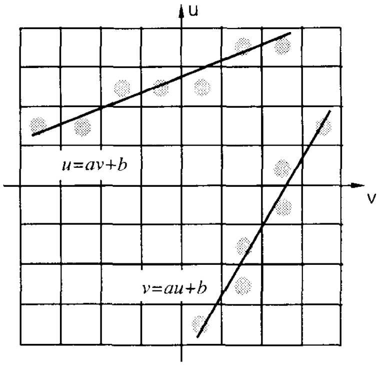


Fig. 5. Line fitting and line equations.

with error covariance matrix

$$\Gamma_{\hat{\boldsymbol{m}}} \triangleq \mathbb{E}(\hat{\boldsymbol{m}} - \boldsymbol{m})(\hat{\boldsymbol{m}} - \boldsymbol{m})^T = \left(A^T \Gamma_y^{-1} A\right)^{-1}. \quad (4.10)$$

If, in addition, $\boldsymbol{\delta}_y$ is normally distributed, the linear unbiased minimum variance estimator of $\boldsymbol{m}$ is the minimum variance estimator among all the unbiased estimators (absolute best and not limited in a class of linear estimators). For proofs, see, e.g., [11], [20], [10].

To estimate our line parameter vector $\boldsymbol{a}$ using this result, we need to know the covariance matrix of $\boldsymbol{\delta}$ in (4.5). The component of $\boldsymbol{\delta}$ is the error of the detected edge pixel location in the direction orthogonal to that of the index variable. As an approximation, we assume that this type of error is unbiased and uncorrelated. Although this assumption is not exactly true even in the case of digitization error, we expect the correlation is not so significant that the accuracy of estimates derived from this assumption is not considerably degraded. We will return to this point in the discussion of our simulation, in which we will see that the accuracy of the estimates does not change much if the actual (correlated) digitization noise is replaced by uncorrelated noise. Using the condition $\Gamma_y = \sigma^2 I$ and the Gauss-Markov theorem, the minimum variance estimator of $\boldsymbol{a}$ is

$$\boldsymbol{a}^* = \left(A^T A\right)^{-1} A^T \boldsymbol{V} \quad (4.11)$$

where the covariance matrix of the error of $\boldsymbol{a}^*$ is given by

$$\Gamma_{\boldsymbol{a}^*} = \sigma^2 \left(A^T A\right)^{-1} = \sigma^2 \begin{bmatrix} \sum_i u_i^2 & \sum_i u_i \\ \sum_i u_i & n \end{bmatrix}^{-1}. \quad (4.12)$$

B. Optimal Solution

Now, we discuss how to determine the unknown motion parameter vector $\boldsymbol{m}$ from the lines that are determined by the least squares fitting. Let $\boldsymbol{a}^*_{ij}$ be the observed line parameter vector of the ith line in jth image frame and $\boldsymbol{a}_{ij}(\boldsymbol{m})$ be the corresponding computed line parameter vector from the motion parameter $\boldsymbol{m}$ (the structure $\boldsymbol{x}$ can be optimally determined from $\boldsymbol{m}$ and be suppressed from $\boldsymbol{a}^*_{ij}(\boldsymbol{m}, \boldsymbol{x})$). A nonlinear extension of the Gauss-Markov theorem leads to minimizing

$$\sum_{i=1}^{n} \sum_{j=1}^{3} \left(\boldsymbol{a}^*_{ij} - \boldsymbol{a}_{ij}(\boldsymbol{m})\right) \Gamma^{-1}_{\boldsymbol{a}^*} \left(\boldsymbol{a}^*_{ij} - \boldsymbol{a}_{ij}(\boldsymbol{m})\right). \quad (4.13)$$

In other words, we want to minimize the discrepancies between the measured observables and the inferred observables. Such an objective function can be directly extend to the use of other types of features. According to the Gauss-Markov theorem, the discrepancies to be minimized should be weighted by the inverse of the covariance matrix of the expected errors in the corresponding measurements. For example, if lines and points are used as matching features, the objective function to be minimized is the sum of the weighted line discrepancies shown in (4.13) and the weighted point discrepancies discussed in [26].

The minimization of nonlinear function (4.13) can be solved iteratively from a good initial guess. The closed-form solution of the linear algorithm discussed above can be used as an initial guess solution for the nonlinear optimization here.

C. How Accurate It Can Possibly Be

With a solution based on noise-corrupted data, an important question to ask is how close the error is from the minimum error allowed by the information in the data. Since the noise in the data is random, this question should be investigated in terms of statistics. There exist theoretical bounds for the covariance matrix of any estimator. The Cramér-Rao bound is one of them.

1) Cramér-Rao Bound [5], [18]: Suppose $\boldsymbol{m}$ is a parameter of probability density $p(\boldsymbol{y}, \boldsymbol{m})$. $\hat{\boldsymbol{m}}$ is an estimator of $\boldsymbol{m}$ based on measurement $\boldsymbol{y}$ with $\mathbb{E}\hat{\boldsymbol{m}} = \boldsymbol{b}(\boldsymbol{m})$. Let $\boldsymbol{z}^T = \frac{\partial \ln p(\boldsymbol{z}, \boldsymbol{m})}{\partial \boldsymbol{m}}$. Define

$$F = \mathbb{E}\boldsymbol{z}\boldsymbol{z}^T. \quad (4.14)$$

The matrix F is called the Fisher information matrix. Let

$$B = \frac{\partial \boldsymbol{b}(\boldsymbol{m})}{\partial \boldsymbol{m}}. \quad (4.15)$$

Then

$$\mathbb{E}(\hat{\boldsymbol{m}} - \boldsymbol{b}(\boldsymbol{m}))(\hat{\boldsymbol{m}} - \boldsymbol{b}(\boldsymbol{m}))^T \geq B F^{\dagger} B^T \quad (4.16)$$

where inequality means that the difference of two sides is nonnegative definite. $F^{\dagger}$ is the pseudo-inverse of F.

Letting $B = I$, the Cramér-Rao bound provides a lower bound for the expected errors of any unbiased estimator. Therefore, we can compare the expected errors with that of a "best possible" unbiased estimator using the Cramér-Rao bound. The evaluation of the Cramér-Rao bound requires noise distribution. With Gaussian noise, the Cramér-Rao bound takes a simpler form.

For the problem investigated here, $\boldsymbol{m}$ consists of the independent motion parameters (three in R, three in S, five in the normalized (unit) version of $(\boldsymbol{T}, \boldsymbol{U})$. The structure $\boldsymbol{x}$ can be estimated from the motion parameter vector $\boldsymbol{m}$ by the methods in step 4) of the algorithm presented in the previous section. The observation vector $\boldsymbol{y}$ is a vector that consists a vectors $\boldsymbol{a}$'s in (4.5) for every line in every image. Therefore, $\boldsymbol{y}$ is a

$6n$-dimensional vector for n line correspondences (n lines in each image). Given any motion parameter vector $\boldsymbol{m}$, we can compute its corresponding structure $\boldsymbol{x}$ and the vector $\boldsymbol{y}_0(\boldsymbol{m})$. Suppose the observation vector is given by additive Gaussian noise: $\boldsymbol{y} = \boldsymbol{y}_0(\boldsymbol{m}) + \boldsymbol{\delta}$, where $\boldsymbol{\delta}$ is a Gaussian noise vector with a zero mean and a block diagonal covariance matrix C with each 2 by 2 block given by (4.12). Then, $p(\boldsymbol{y}, \boldsymbol{m})$ is a $6n$-dimensional distribution with mean $\boldsymbol{y}_0(\boldsymbol{m})$ and the same block diagonal covariance matrix C. The Cramér-Rao bound for any unbiased estimator gives

$$\mathrm{E}(\hat{\boldsymbol{m}} - \boldsymbol{m})(\hat{\boldsymbol{m}} - \boldsymbol{m})^T \geq \left(\left(\frac{\partial \boldsymbol{y}_0(\boldsymbol{m})}{\partial \boldsymbol{m}} \right) C \left(\frac{\partial \boldsymbol{y}_0(\boldsymbol{m})}{\partial \boldsymbol{m}} \right)^T \right)^{-1}. \tag{4.17}$$

For further discussion, see, e.g., [3], [27].

V. Simulations

Simulations have been performed to demonstrate correctness of the algorithm as well as sensitivity of the solution to noise.

A. Setup

The focal length of the camera is one unit, and the image is an $s \times s$ square. Lines are generated randomly for time t_0. The centers of the lines are uniformly distributed between depths 5 and 15. The orientation of the lines are uniformly distributed over all directions. The length of the lines is uniformly distributed in $[4s, 8s]$, and only the visible part of the lines are used for line fitting. (Usually, a line following procedure will detect connected edge points along the line. However, a small portion of missing edge points does not significantly degrade the accuracy of the measured line.) The lines are moved to another position at time t_1 by a rotation represented by matrix R and then a translation represented by translation vector $\boldsymbol{T}$. Similarly, at time t_2, the lines are moved from t_0 by a rotation and then a translation represented by S and $\boldsymbol{U}$, respectively.

The lines at each time are projected onto the image plane, and the edge points are corrupted by digitization noise: For a 256×256 image, there are 256×256 pixels, and the image coordinates of edge points have 256 evenly spaced levels for u and v coordinates, respectively. The real image coordinates of edge points are rounded off to the closest levels. Different resolution may be used to simulate a different amount of noise. To take into account common line finding processes, the line position is determined by a least squares fitting to the visible edge points (pixel centers) of the line. As we discussed, the relatively higher accuracy of the line position obtained by such a line fitting is one of the main advantages for using lines as matching primitives.

As we know, the error of a detected line may arise from many different sources, and the statistical nature of the error depends very much on the line detector that is actually used. The simulated noise here is not meant to indicate what will actually happen in practice since the wide variety of line detectors makes such an attempt impossible and unnecessary. An exhaustive or complex noise model may contain many noise parameters (e.g., bias and correlation), whose values are difficult to select and evaluate in terms of their practical applicability. The noise model here is meant to be simple enough so that the performance of the algorithms can be examined and evaluated in a clear way.

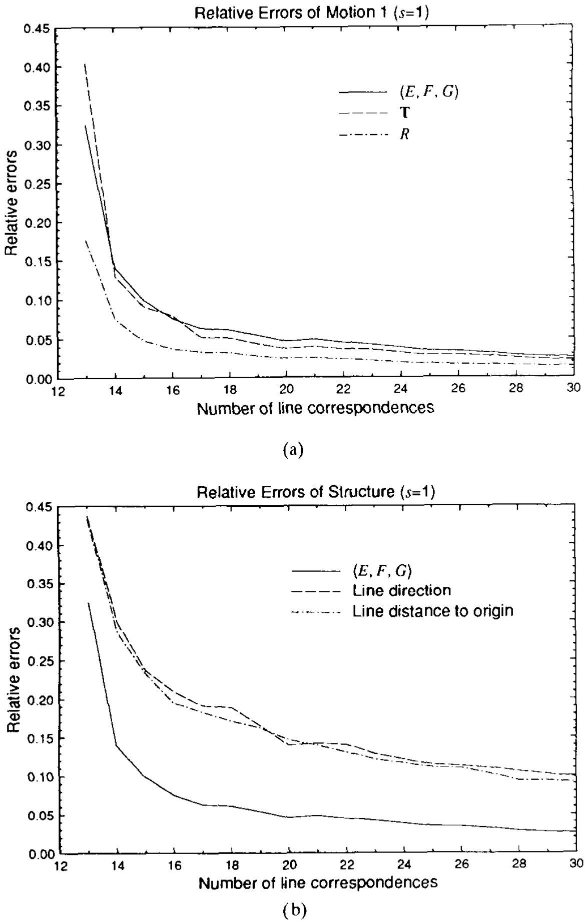
The errors shown in the following are all relative errors. The relative error of a matrix (vector is a column matrix) is defined by the Euclidean norm (square root of the sum of squared elements) of the error matrix (difference between the estimated one and the true one) divided by the Euclidean norm of the true matrix.

B. Linear Algorithm

In noise-free cases, the errors in the solution given by the algorithm are mainly caused by computer round-off errors. Sun workstations with double precision were used for the simulations. The relative errors in the solution were of the order of 10^{-10} if edge points were not rounded off to pixel centers.

In the presence of noise, the errors in the solutions depend on the configuration of the lines. To show the general sensitivity to the noise, the errors are averaged through 100 random trials, where each trial has the same motion but a different set of randomly generated lines. The noise is simulated by the pixel round-off error with different image resolutions. Simulations of our line-based algorithm and the results of error estimation discussed in [25] and [28] showed that the errors in the solutions are roughly proportional to the amount of noise in the image plane. For example, if the noise in image plane is doubled, the errors in solutions are also doubled. For the data shown here, the image resolution used is 256×256.

Since the errors in the solutions are very sensitive to the field of view, we show the results using two image sizes $s = 1$ and $s = 0.7$ (note: focal length is 1). The results of $s = 1$ are shown in Fig. 6. The motion parameters are as follows: R corresponds to a rotation about axis (1, 1, 1) by an angle of 6°, and S corresponds to a rotation about axis (0, 1, -1) by an angle of 5°. $\boldsymbol{T} = (2, -2, 2)$, and $\boldsymbol{U} = (-1, 2 - 2)$. Fig. 6(a) shows the average relative errors of the intermediate parameters (E, F, G), R, and $\boldsymbol{T}$, versus the number of line correspondences. The average relative errors of S, and $\boldsymbol{U}$ are virtually the same as those of R and $\boldsymbol{T}$ and are omitted. We can see that with a minimal 13 lines, the errors are relatively large. With a minimal number of line correspondences, no overdetermination is available in the set of equations of (2.13). Therefore, the intermediate parameters cannot be determined accurately. Furthermore, some short lines in the images may give very unreliable projection normals. After a few lines are added to the minimally required 13, the errors decrease significantly. (It seems that far more than 13 pairs of lines can be matched from real world images [9], [16]. We may expect that overdetermination over 13 line correspondences is generally available in practice.) The location of a line can be specified by the direction of the line ($\hat{\boldsymbol{l}}$), the projection plane passing through the line and the origin (seen by the camera as an observable), and the distance from the line to the origin ($\|\boldsymbol{x}_p\|$). Fig. 6(b) shows the average errors of the recovered



(a)

(b)

Fig. 6. Average relative errors of for $s = 1$ (image size: $s \times s$) through 100 random trials: (a) Motion parameters of motion 1; (b) structure.

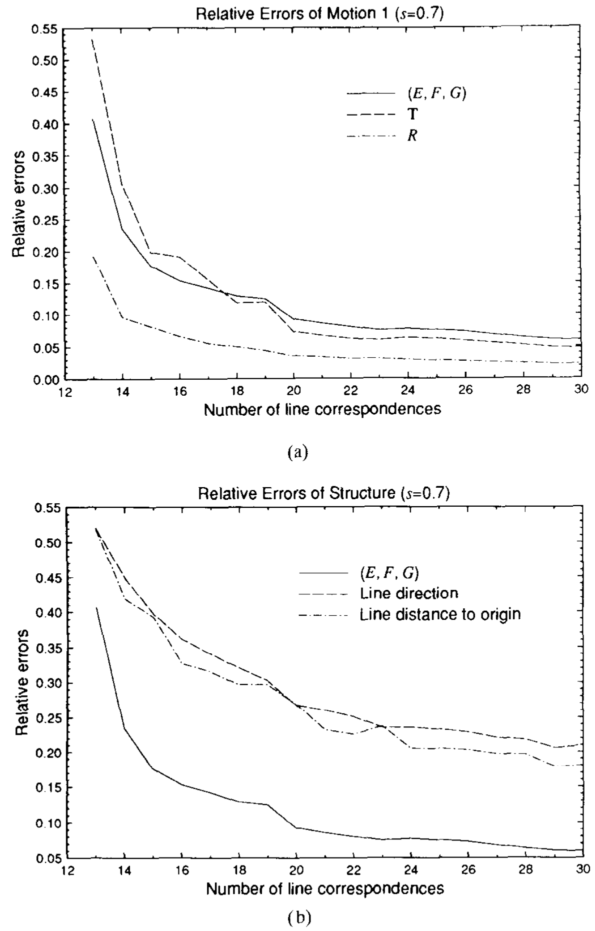
direction of lines $\ell$and those of recovered relative errors of the distance from the line to the origin.

Fig. 7 shows the relative errors for $s = 0.7$. The rotation parameters are the same as those for $s = 1$ in Fig. 6. The translations are modified accordingly to account for the reduced field of view: $\boldsymbol{T} = (1.5, -1.5, 2)$, and $\boldsymbol{U} = (-1, 1.5, -2)$. (The errors in the estimates of motion 2 are very similar to the corresponding ones of motion 1, and therefore, the corresponding plots are omitted from Figs. 7 to 9.) As can be seen, the errors in this case are considerably larger than the corresponding errors with a larger field of view (Fig. 6).

As indicated in our simulation, the error in the solution is roughly inversely proportional to the magnitude of the translations. Therefore, as translation vanishes the error increases quickly. The shape of such an increase is very similar to that of the point-based algorithm (see, e.g., Fig. 10 in [28]).

C. Algorithm with Optimization

We next present the results with optimization. In actual implementation, to obtain a better initial solution for minimizing (4.13), we first minimize the equation error of (2.33) starting from the solution of the linear algorithm. Then, the resulting parameters are used as an initial guess solution to minimize (4.13). The errors of the final optimal solutions are shown in Fig. 8 together with those of the initial guesses shown in Fig. 6. Very significant improvements over those of the linear algorithm are achieved. The results of minimizing equation errors in (2.33) and those of minimizing (4.13) are shown in Fig. 9. As can be seen, although minimizing equation errors of (2.33) is not as good as minimizing (4.13), the difference is not very large. Therefore, when an exact optimal solution is not necessary, minimizing the simpler expression (2.33) may suffice.



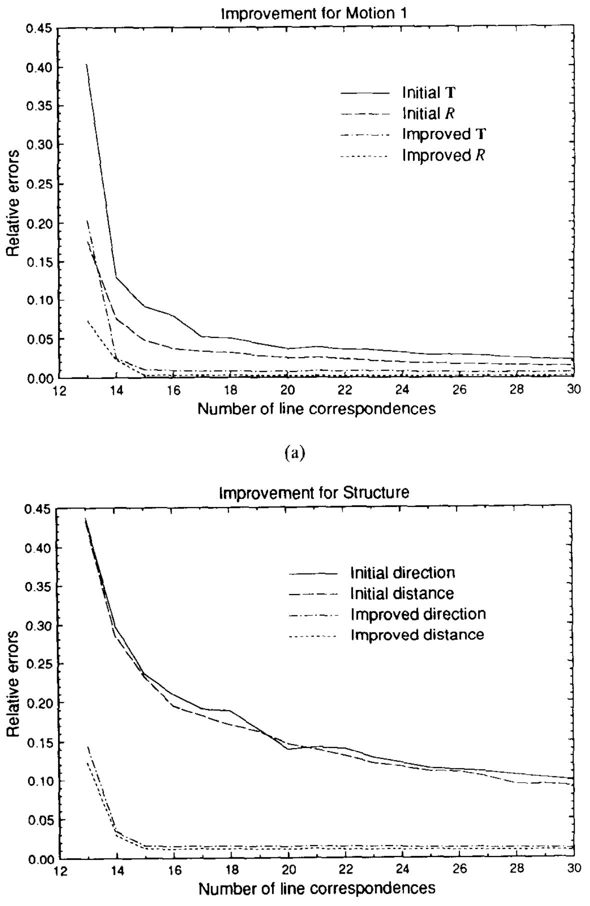
(a)

(b)

Fig. 7. Average relative errors of for $s = 0.7$ (image size: $s \times s$) through 100 random trials: (a) Motion parameters of motion 1; (b) structure.

D. Compared with the Bound

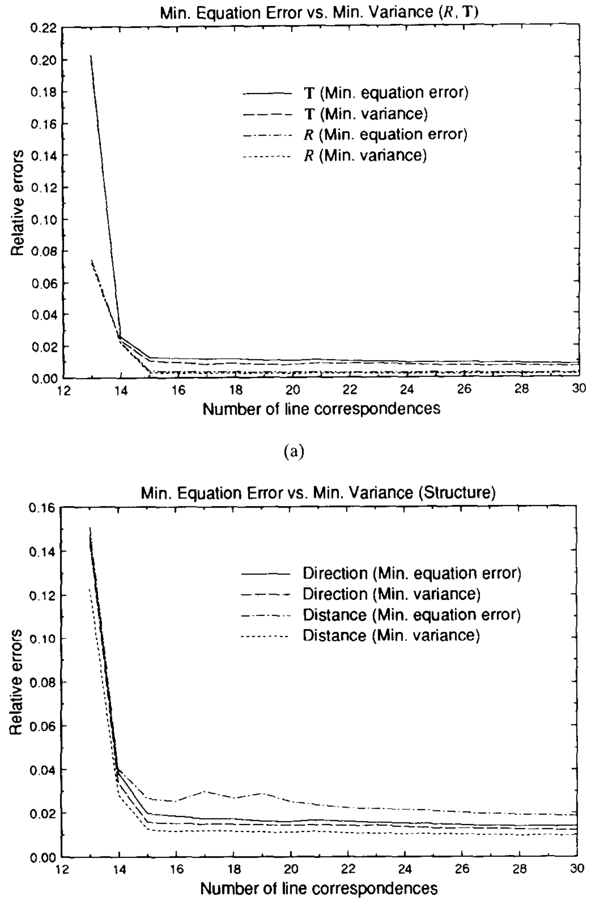
In Fig. 10, the Cramér-Rao lower bound is shown together with the actual errors. The setup and the motion parameters are the same as those in Fig. 6, and zero mean, uncorrelated Gaussian noise with a variance equal to that of digitization noise of 256 × 256 images is added to edge points. The error bounds are very close to the actual errors, except for the small number of line correspondences. With a small number of lines,



(a)

(b)

Fig. 8. Improvement of optimization over the linear algorithm versus number of line correspondences. 100 random trials: (a) Motion parameters of motion 1; (b) structure.



(a)

(b)

Fig. 9. Minimizing equation error (2.33) versus minimizing variance (4.13) versus number of line correspondences. 100 random trials: (a) Motion parameters of motion 1; (b) structure.

the errors in the initial guesses are relatively large (see Figs. 5 and 6), and therefore, the true optimal solutions cannot always be reliably obtained. However, as shown in Fig. 10, with a few extra lines in addition to a minimal 13, the linear algorithm followed by optimization essentially reached the Cramér-Rao lower bound, which is the lower error bound of a best possible unbiased estimator.

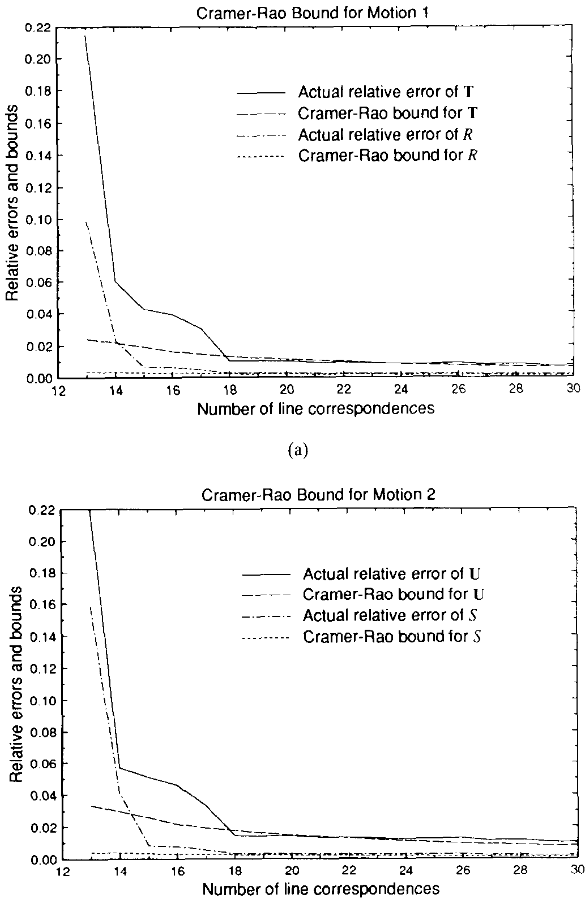
A point is worth mentioning here. The noise added in Fig. 10 is uncorrelated Gaussian noise, whereas in Figs. 6 to 9, the noise is actual spatial digitization noise, which is not exactly uncorrelated. In (4.11), we have assumed that the digitization noise is uncorrelated, which will potentially degrade the performance. However, a comparison between Figs. 8 and 10 for the improved results tells us that the corresponding errors are very similar. This seems to indicate that the assumption that the digitization noise is uncorrelated does not significantly affect the performance of the algorithm.

VI. Conclusions and Discussions

A new linear algorithm is presented for estimating motion and structure parameters from line correspondences. Relatively compact computational schemes are derived in order to avoid, as much as possible, spurious solutions and degenerate cases and to make use of the redundancy in the data. The uniqueness of the solution has been established. As long as the coefficient matrix of the linear equations (2.34) is not degenerate, the algorithm gives a unique solution to the motion parameters. Some necessary and sufficient conditions for the lines to result in a degenerate coefficient matrix are presented.

An approach to optimal estimation of motion and structure from line correspondences has also been introduced. The reliability of each measured line is represented by an error covariance matrix, which is utilized for an optimal solution. In order to reliably reach the global minimal point of the nonlinear objective function, the closed-form solution is used as an initial guess solution.

From the results of our simulations, it appears that the accuracy of the solutions by our optimal line-based algorithm is close to that of the corresponding optimal point-based algorithm [26], with the same amount of image plane noise, the same number of line or point correspondences, and a similar



(a)

(b)

Fig. 10. Actual errors, Cramér-Rao bound for Gaussian noise versus number of line correspondences. 100 random trails: (a) Motion parameters of motion 1; (b) motion parameters of motion 2.

amount of motion. A line-to-line correspondence provides only one component of the image plane displacement (along the normal to the line), whereas a point-to-point correspondence provides both components. Therefore, in some sense, line-to-line correspondences contain less information than point-to-point correspondences. How can one expect the line-based algorithm to perform as well as point-based algorithms? The key is that a line-based algorithm can use the redundancy in the edge points to obtain more accurate measurement of line positions. In other words, a line fitting step used in our algorithm is very important for line-based algorithms.

Since the accuracy of the optimal solutions is close to the Cramér-Rao lower error bound for any unbiased estimator, the obtained performance appears to leave little improvement beyond.

Appendix A

Proof of Theorem 1: If $\boldsymbol{R}_1/\boldsymbol{T}$, Lemma 2 presented in Appendix B concludes that $rank(E) \leq 1$. Otherwise, $E^T\boldsymbol{V}_1 = \boldsymbol{0}$, where $\boldsymbol{V}_1 = \boldsymbol{R}_1 \times \boldsymbol{T} \neq \boldsymbol{0}$. Thus, $rank(E) \leq 2$. Similarly $rank(F) \leq 2$, and $rank(G) \leq 2$.

Case 1: Since the ranks of E, F, G are all equal to 2, from Lemma 2, $\boldsymbol{T}$ is not parallel to any column vectors of R, and $\boldsymbol{U}$ is not parallel to any column vectors of S. Thus, $\boldsymbol{V}_i//\boldsymbol{T} \times \boldsymbol{R}_i$, $i = 1, 2, 3$. $rank(A) = 2$ if $rank(M) = 2$, where

$$M = \left[\hat{\boldsymbol{T}}\times \boldsymbol{R}_1 \quad \hat{\boldsymbol{T}}\times \boldsymbol{R}_2 \quad \hat{\boldsymbol{T}}\times \boldsymbol{R}_3\right] = \left[\hat{\boldsymbol{T}}\right]_\times R \tag{A.1}$$

and $\hat{\boldsymbol{T}} = \|\boldsymbol{T}\|^{-1}\boldsymbol{T}$. We prove that $rank(M) = 2$. In fact, let the unit vectors $\hat{\boldsymbol{T}}$, $\hat{\boldsymbol{T}}_2$, and $\hat{\boldsymbol{T}}_3$ be such that $Q = \left[\hat{\boldsymbol{T}}\hat{\boldsymbol{Q}}_2\ \hat{\boldsymbol{Q}}_3\right]$ is an orthonormal 3×3 matrix. $G = R^TQ$ is also orthonormal. Post-multiplying the two sides of (A.1) by G, we get

$$MG = \left[\hat{\boldsymbol{T}}\right]_\times RG = \left[\hat{\boldsymbol{T}}\right]_\times Q = \left[0 \quad \hat{\boldsymbol{T}}\times\hat{\boldsymbol{Q}}_2 \quad \hat{\boldsymbol{T}}\times\hat{\boldsymbol{Q}}_3\right].$$

We see that the second and the third columns of MG are orthonormal from the definition of Q. Therefore, $rank(M) = rank(MG) = 2$.

Case 2: We need to prove that if $rank(A) \leq 1$, $\boldsymbol{T}$ is essentially determined by $\boldsymbol{T}//(\boldsymbol{E}_i \times \boldsymbol{V}_2) \times \boldsymbol{V}_2$, and it is true that $(\boldsymbol{E}_i \times \boldsymbol{V}_2) \times \boldsymbol{V}_2 \neq \boldsymbol{0}$. Since $\boldsymbol{V}_i//\boldsymbol{T} \times \boldsymbol{R}_i$, $i = 2, 3$, $rank(A) \leq 1$ implies

$$\boldsymbol{0} = (\boldsymbol{T} \times \boldsymbol{R}_2) \times (\boldsymbol{T} \times \boldsymbol{R}_3) = ((\boldsymbol{T} \times \boldsymbol{R}_2) \cdot \boldsymbol{R}_3)\boldsymbol{T} - 0\boldsymbol{R}_3$$

where the last equation follows by using the identity $\boldsymbol{a} \times (\boldsymbol{b} \times \boldsymbol{c}) = (\boldsymbol{a} \cdot \boldsymbol{c})\boldsymbol{b} - (\boldsymbol{a} \cdot \boldsymbol{b})\boldsymbol{c}$. Thus, $(\boldsymbol{T} \times \boldsymbol{R}_2) \cdot \boldsymbol{R}_3 = 0$, i.e., $\boldsymbol{T}$, $\boldsymbol{R}_2$, $\boldsymbol{R}_3$ are coplanar. Therefore

$$\boldsymbol{T} \cdot \boldsymbol{R}_1 = 0 \tag{A.2}$$

and $\boldsymbol{V}_2//\boldsymbol{T} \times \boldsymbol{R}_2//\boldsymbol{R}_1$. Thus, $\boldsymbol{V}_2 = \pm\boldsymbol{R}$. Since $rank(E) \leq 1$ and $\boldsymbol{T} \cdot \boldsymbol{R}_1 = 0$, we get $\boldsymbol{S}_1//\boldsymbol{U}$ from Lemma 1. Let $\boldsymbol{U} = k\boldsymbol{S}_1$ for some real number k. We have $E = (k\boldsymbol{R}_1 - \boldsymbol{T})\boldsymbol{S}_1^T$ and $\boldsymbol{E}_i = l(k\boldsymbol{R}_1 - \boldsymbol{T})$ for some real number $l \neq 0$ since $\boldsymbol{S}_1 \neq \boldsymbol{0}$. Therefore, using (A.2) yields

$$\begin{aligned}(\boldsymbol{E}_i \times \boldsymbol{V}_2) \times \boldsymbol{V}_2 &= l((k\boldsymbol{R}_1 - \boldsymbol{T}) \times \boldsymbol{R}_1) \times \boldsymbol{R}_1 \\ &= -l(\boldsymbol{T} \times \boldsymbol{R}_1) \times \boldsymbol{R}_1 = l\boldsymbol{T}.\end{aligned}$$

$\boldsymbol{T} \neq \boldsymbol{0}$ and $l \neq 0$ give $(\boldsymbol{E}_i \times \boldsymbol{V}_2) \times \boldsymbol{V}_2 \neq \boldsymbol{0}$.

Case 3: From Lemma 1, we have

$$\boldsymbol{T}//\boldsymbol{R}_1 \quad \text{or} \quad \boldsymbol{T}//\boldsymbol{R}_2 \tag{A.3}$$

and

$$\boldsymbol{U}//\boldsymbol{S}_1 \quad \text{or} \quad \boldsymbol{U}//\boldsymbol{S}_2. \tag{A.4}$$

Since $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ are orthogonal vectors, as are $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$, (A.3) and (A.4) give only two possible combinations: 1) $\boldsymbol{T}//\boldsymbol{R}_1$ and $\boldsymbol{U}//\boldsymbol{S}_2$; 2) $\boldsymbol{T}//\boldsymbol{R}_2$ and $\boldsymbol{U}//\boldsymbol{S}_2$.

For 1); letting $\boldsymbol{T} = k\boldsymbol{R}_1$, $(k \neq 0)$, and $\boldsymbol{U} = l\boldsymbol{S}_2$, $(l \neq 0)$ yields

$$E = \boldsymbol{R}_1(\boldsymbol{U} - k\boldsymbol{S}_1)^T$$

and

$$F = (l\boldsymbol{R}_2 - \boldsymbol{T})\boldsymbol{S}_2^T = (l\boldsymbol{R}_2 - k\boldsymbol{R}_1)\boldsymbol{S}_2^T$$

which gives

$$\boldsymbol{V}_{1a} \times \boldsymbol{V}_{1b} // \boldsymbol{R}_1 \tag{A.5}$$

and

$$\boldsymbol{V}_{2a} \times \boldsymbol{V}_{2b} // (l\boldsymbol{R}_2 - k\boldsymbol{R}_1).$$

On the other hand, $\boldsymbol{V}_3 // \boldsymbol{T} \times \boldsymbol{R}_3 = k\boldsymbol{R}_1 \times \boldsymbol{R}_3 = k\boldsymbol{R}_2$. Therefore

$$\boldsymbol{V}_3 \cdot (\boldsymbol{V}_{1a} \times \boldsymbol{V}_{1b}) = 0$$
$$\boldsymbol{V}_3 \cdot (\boldsymbol{V}_{2a} \times \boldsymbol{V}_{2b}) \neq 0.$$

Equation (A.5) gives

$$\boldsymbol{T} // \boldsymbol{R}_1 // \boldsymbol{V}_{1a} \times \boldsymbol{V}_{1b}.$$

For 2), similarly, we have

$$\boldsymbol{V}_3 \cdot (\boldsymbol{V}_{1a} \times \boldsymbol{V}_{1b}) \neq 0$$
$$\boldsymbol{V}_3 \cdot (\boldsymbol{V}_{2a} \times \boldsymbol{V}_{2b}) = 0$$

and

$$\boldsymbol{T} // \boldsymbol{R}_2 // \boldsymbol{V}_{2a} \times \boldsymbol{V}_{2b}.$$

□

Appendix B

Lemma 1: Let $\boldsymbol{R}_1$, $\boldsymbol{S}_1$, $\boldsymbol{T}$, and $\boldsymbol{U}$ be nonzero vectors, and

$$E = \boldsymbol{R}_1 \boldsymbol{U}^T - \boldsymbol{T}\boldsymbol{S}_1^T.$$

Then

1) If $rank(E) = 0$, then $\boldsymbol{R}_1 // \boldsymbol{T}$ and $\boldsymbol{S}_1 // \boldsymbol{U}$.
2) If $rank(E) = 1$, then $\boldsymbol{R}_1 // \boldsymbol{T}$ or $\boldsymbol{S}_1 // \boldsymbol{U}$.
3) If $\boldsymbol{R}_1 // \boldsymbol{T}$, or $\boldsymbol{U} // \boldsymbol{S}_1$, then $rank(E) \leq 1$.

Proof: $rank(E) = 0$ implies $\boldsymbol{R}_1 \boldsymbol{U}^T = \boldsymbol{T}\boldsymbol{S}_1^T$, the conclusion of 1) immediately follows.

2) $rank(E) = 1$ implies that there exist two nonzero vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ such that

$$E = \boldsymbol{R}_1 \boldsymbol{U}^T - \boldsymbol{T}\boldsymbol{S}_1^T = \boldsymbol{a}\boldsymbol{b}^T. \tag{A.6}$$

Let $\boldsymbol{b}$, $\boldsymbol{b}_1$, $\boldsymbol{b}_2$ be nonzero vectors, and they are mutually orthogonal. Post-multiplying both sides of (A.6) by $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ yields

$$\boldsymbol{R}_1(\boldsymbol{U} \cdot \boldsymbol{b}_1) - \boldsymbol{T}(\boldsymbol{S}_1 \cdot \boldsymbol{b}_1) = \boldsymbol{0}$$
$$\boldsymbol{R}_1(\boldsymbol{U} \cdot \boldsymbol{b}_2) - \boldsymbol{T}(\boldsymbol{S}_1 \cdot \boldsymbol{b}_2) = \boldsymbol{0}.$$

If $\boldsymbol{R}_1 \times \boldsymbol{T} \neq 0$, we have

$$\boldsymbol{U} \cdot \boldsymbol{b}_1 = \boldsymbol{U} \cdot \boldsymbol{b}_2 = \boldsymbol{S}_1 \cdot \boldsymbol{b}_1 = \boldsymbol{S}_2 \cdot \boldsymbol{b}_2 = 0$$

which implies $\boldsymbol{U} // \boldsymbol{b}$ and $\boldsymbol{S}_1 // \boldsymbol{b}$ and therefore, $\boldsymbol{U} // \boldsymbol{S}_1$.

3) Let $\boldsymbol{R}_1 // \boldsymbol{T}$. There exists a number k such that $\boldsymbol{T} = k\boldsymbol{R}_1$. Thus, $E = \boldsymbol{R}_1(\boldsymbol{U} - k\boldsymbol{S}_1)^T$, which implies $rank(E) \leq 1$. Similarly, the case $\boldsymbol{S}_1 // \boldsymbol{U}$ can be proved. □

Appendix C

Proof of Lemma 2: Since there exists at least one set of solutions, i.e., the true one, we prove that solutions for two combinations of (s_1, s_2) yield contradiction. Assume $(s_1, s_2) = (\dot{s}_1, \dot{s}_2)$ is the correct assignment, which gives the solutions

$$(R, S) = (R_p, S_p) \tag{A.7}$$

whcih correspond to

$$\boldsymbol{T} = \dot{s}_1 \|\boldsymbol{T}\| \hat{\boldsymbol{T}}_s, \qquad \boldsymbol{U} = \dot{s}_2 \|\boldsymbol{U}\| \hat{\boldsymbol{U}}_s. \tag{A.8}$$

Reversing the sign for s_1 yields R_n in (2.22), and reversing the sign for s_2 yields S_n in (2.24). From (2.22), we have

$$\left[\hat{\boldsymbol{T}}_s\right]_\times R_p = -\left[\hat{\boldsymbol{T}}_s\right]_\times R_n.$$

From Lemma 3 presented in Appendix D, we have

$$R_n = R(\boldsymbol{T}, \pi) R_p \tag{A.9}$$

where $R(\boldsymbol{T}, \pi)$ is the rotation matrix representing the rotation about vector $\boldsymbol{T}$ by an angle π. Similarly, we have

$$S_n = S(\boldsymbol{U}, \pi) S_p. \tag{A.10}$$

We first prove that it is impossible that for both $(\pm\dot{s}_1, \dot{s}_2)$, there exist solutions. Otherwise, from the first equation in (2.20), we have

$$E = \dot{s}_1 \|\boldsymbol{U}\| \boldsymbol{R}_{p1} \hat{\boldsymbol{U}}_s^T - \dot{s}_2 \|\boldsymbol{T}\| \hat{\boldsymbol{T}}_s \boldsymbol{S}_{p1}^T$$

and

$$E = -\dot{s}_1 \|\boldsymbol{U}\| \boldsymbol{R}_{n1} \hat{\boldsymbol{U}}_s^T - \dot{s}_2 \|\boldsymbol{T}\| \hat{\boldsymbol{T}}_s \boldsymbol{S}_{p1}^T$$

where the subscripts denote the corresponding columns of the rotation matrix. Subtracting both sides yields

$$\|\boldsymbol{U}\| \boldsymbol{R}_{p1} \hat{\boldsymbol{U}}_s^T = -\|\boldsymbol{U}\| \boldsymbol{R}_{n1} \hat{\boldsymbol{U}}_s^T.$$

Since $\boldsymbol{U} \neq \boldsymbol{0}$, we get

$$\boldsymbol{R}_{p1} = -\boldsymbol{R}_{n1}.$$

From the remaining two equations of (2.20), we get the similar results for the other two columns of the rotation matrices R_p and S_n. Therefore

$$R_p = -R_n.$$

This is a contradiction because R_p and R_n are both rotation matrices whose determinants are equal to one.

Similarly $(\dot{s}_1, \pm\dot{s}_2)$ cannot both have solutions.

The only other possible assignment that has a solution is $(-\dot{s}_1, -\dot{s}_2)$. Suppose it has a solution. For $i = 1, 2, 3$, the equations of (2.20) give

$$E = \dot{s}_1 \|\boldsymbol{U}\| \boldsymbol{R}_{pi} \hat{\boldsymbol{U}}_s^T - \dot{s}_2 \|\boldsymbol{T}\| \hat{\boldsymbol{T}}_s \boldsymbol{S}_{pi}^T$$

and

$$E = -\dot{s}_1 \|\boldsymbol{U}\| \boldsymbol{R}_{ni} \hat{\boldsymbol{U}}_s^T + \dot{s}_2 \|\boldsymbol{T}\| \hat{\boldsymbol{T}}_s \boldsymbol{S}_{ni}^T.$$

These two equations yield

$$\dot{s}_1\|\boldsymbol{U}\|(\boldsymbol{R}_{pi}+\boldsymbol{R}_{ni})\hat{\boldsymbol{U}}_s^T = \dot{s}_2\|\boldsymbol{T}\|\hat{\boldsymbol{T}}_s\left(\boldsymbol{S}_{pi}^T+\boldsymbol{S}_{ni}^T\right). \qquad \text{(A.11)}$$

From (A.9)–(A.11), we have

$$\|\boldsymbol{U}\|\left(R\left(\hat{\boldsymbol{T}},\pi\right)+I\right)\boldsymbol{R}_{pi})\hat{\boldsymbol{U}}_s^T = \|\boldsymbol{T}\|\hat{\boldsymbol{T}}_s\left(S\left(\hat{\boldsymbol{U}},\pi\right)+I\right)\boldsymbol{S}_{pi})^T. \qquad \text{(A.12)}$$

Since

$$\hat{\boldsymbol{T}}_s^T R\left(\hat{\boldsymbol{T}},\pi\right) = \hat{\boldsymbol{T}}_s$$

premultiplying both sides of (A.12) by $\hat{\boldsymbol{T}}_s^T$ and post-multiplying the result by $\hat{\boldsymbol{U}}_s$ give

$$2\dot{s}_1\|\boldsymbol{U}\|\hat{\boldsymbol{T}}_s^T\boldsymbol{R}_{pi} = 2\dot{s}_2\|\boldsymbol{T}\|\boldsymbol{S}_{pi}^T\hat{\boldsymbol{U}}_s$$

where $i = 1,2,3$. Therefore

$$2\dot{s}_1\|\boldsymbol{U}\|\hat{\boldsymbol{T}}_s^T R_p = 2\dot{s}_2\|\boldsymbol{T}\|\hat{\boldsymbol{U}}_s^T S_p. \qquad \text{(A.13)}$$

Since R and S are rotation matrices, from (A.13), we have $\|\boldsymbol{T}\| = \|\boldsymbol{U}\|$. Then, (A.3.16), (A.7), and (A.8) yield

$$\boldsymbol{T}^T R = \boldsymbol{U}^T S.$$

This is a contradiction to $R^T\boldsymbol{T} \neq S^T\boldsymbol{U}$. □

Appendix D

Lemma 3: For any $\boldsymbol{T} \neq \boldsymbol{0}$

$$[\boldsymbol{T}]_\times R_p = -[\boldsymbol{T}]_\times R_n \qquad \text{(A.14)}$$

yields the relation

$$R_n = R(\boldsymbol{T},\pi)R_p \qquad \text{(A.15)}$$

where $R(\boldsymbol{T},\pi)$ is the rotation matrix representing the rotation about vector $\boldsymbol{T}$ by angle π.

Proof: Let $\hat{\boldsymbol{T}} = \|\boldsymbol{T}\|^{-1}\boldsymbol{T}$, and the matrix $\left[\hat{\boldsymbol{T}}\ \hat{\boldsymbol{T}}_1\ \hat{\boldsymbol{T}}_2\right]$ form a right-handed orthonormal matrix. Equation (A.14) gives

$$R_p^T\left[\hat{\boldsymbol{T}}\right]_\times = -R_n^T\left[\hat{\boldsymbol{T}}\right]_\times. \qquad \text{(A.16)}$$

Since $rank\left(\left[\hat{\boldsymbol{T}}\right]_\times\right) = 2$, and $\left[\hat{\boldsymbol{T}}\right]_\times\hat{\boldsymbol{T}} = \boldsymbol{0}$, the columns of $\left[\hat{\boldsymbol{T}}\right]_\times$ and $\hat{\boldsymbol{T}}$ span the 3-D space R^3. Since $\boldsymbol{T}_1$ and $\hat{\boldsymbol{T}}_2$ are both orthogonal to $\hat{\boldsymbol{T}}$, they can be represented by the linear combination of the columns of $\left[\hat{\boldsymbol{T}}\right]_\times : \boldsymbol{T}_i = \left[\hat{\boldsymbol{T}}\right]_\times\boldsymbol{Y}_i, i = 1,2$. From (A.16), we have

$$R_p^T\hat{\boldsymbol{T}}_i = R_p^T\left[\hat{\boldsymbol{T}}\right]_\times\boldsymbol{Y}_i = -R_n^T\left[\hat{\boldsymbol{T}}\right]_\times\boldsymbol{Y}_i = -R_n^T\hat{\boldsymbol{T}}_i$$

$i = 1,2$. Therefore

$$R_nR_p^T\left[\hat{\boldsymbol{T}}\ \ \hat{\boldsymbol{T}}_1\ \ \hat{\boldsymbol{T}}_2\right] = \left[\boldsymbol{X}\ \ -\hat{\boldsymbol{T}}_1\ \ -\hat{\boldsymbol{T}}_2\right] \qquad \text{(A.17)}$$

where $\boldsymbol{X}$ must be equal to $\hat{\boldsymbol{T}}$ since the left-hand side is a rotation matrix. Since

$$\left[\boldsymbol{T}\ \ -\hat{\boldsymbol{T}}_1\ \ -\hat{\boldsymbol{T}}_2\right] = R\left(\hat{\boldsymbol{T}},\pi\right)\left[\boldsymbol{T}\ \ \hat{\boldsymbol{T}}_1\ \ \hat{\boldsymbol{T}}_2\right] \qquad \text{(A.18)}$$

(A.17) and (A.18) give

$$R_nR_p^T = R\left(\hat{\boldsymbol{T}},\pi\right).$$

This yields (A.15) immediately. □

Appendix E

Proof of Theorem 3: From (2.5) and (2.6), we have

$$\boldsymbol{n}_1 = R(\boldsymbol{n}_0 - \boldsymbol{O}_1 \times \boldsymbol{l}) \qquad \text{(A.18)}$$

$$\boldsymbol{n}_2 = S(\boldsymbol{n}_0 - \boldsymbol{O}_2 \times \boldsymbol{l}). \qquad \text{(A.19)}$$

Substituting (A.18) and (A.19) into (2.13) yields (3.4), where we define

$$\begin{aligned}\left(\tilde{E},\tilde{F},\tilde{G}\right) &= \left(R^TES, R^TFS, R^TGS\right)\\ &= \left(P_1 - Q_1^T, P_2 - Q_2^T, P_3 - Q_3^T\right). \qquad \text{(A.20)}\end{aligned}$$

The last equation in (A.20) follows from (2.12) using the fact that R and S are orthonormal.

(E,F,G) in (2.12) satisfies (2.13). From (A.20), trivial $(\tilde{E},\tilde{F},\tilde{G})$ satisfies (3.4). From (A.20), (3.4) has only the trivial solution if and only if (2.13) essentially determines (E,F,G). □

Appendix F

Proof of Corollary: If $\boldsymbol{O}_1 = \boldsymbol{0}$, let $(\tilde{E},\tilde{F},\tilde{G}) = \left(V_1^T, V_2^T, V_3^T\right)$, where V_i is a matrix with the ith column being an arbitrary vector $\boldsymbol{v}_i$, and the other columns are zeroes. Equation (3.4) becomes $[\boldsymbol{n}]_\times\boldsymbol{n}[\boldsymbol{v}_1\ \boldsymbol{v}_2\ \boldsymbol{v}_3]^T\boldsymbol{n}_2 = \boldsymbol{0}$, which holds since $[\boldsymbol{n}]_\times\boldsymbol{n} = \boldsymbol{0}$. This means that (3.4) has nontrivial solutions. Similarly, if $\boldsymbol{O}_2 = \boldsymbol{O}$ (3.4) has nontrivial solutions. Finally, if $\boldsymbol{O}_1 = \boldsymbol{O}_2$, then $\tilde{\boldsymbol{n}}_1 = \tilde{\boldsymbol{n}}_2$. As long as $\tilde{E},\tilde{F},\tilde{G}$ are antisymmetrical matrices, (3.4) holds since $(\boldsymbol{v})^TM\boldsymbol{v} = 0$ holds for any symmetrical matrix M. Therefore, (3.4) has nontrivial solutions. □

Appendix G

Proof of Theorem 4: Obviously, the normal to a plane passing through the line and $\boldsymbol{O}_k$ is equal to $(\boldsymbol{x}_p - \boldsymbol{O}_1) \times \boldsymbol{l}$, $k = 1,2$. Equation (3.4) can be rewritten as

$$[\boldsymbol{n}]_\times\begin{bmatrix}(\tilde{\boldsymbol{n}}_1)^T\tilde{E}\tilde{\boldsymbol{n}}_2\\(\tilde{\boldsymbol{n}}_1)^T\tilde{F}\tilde{\boldsymbol{n}}_2\\(\tilde{\boldsymbol{n}}_1)^T\tilde{G}\tilde{\boldsymbol{n}}_2\end{bmatrix} = \boldsymbol{0}. \qquad \text{(A.21)}$$

Equation (A.21) has at most two independent scalar equations since $rank([\boldsymbol{n}]_\times) = 2$ for $\boldsymbol{n} \neq \boldsymbol{0}$. We need to exclude one equation that is a linear combination of other two. Let $\boldsymbol{a} = (a_1,a_2,a_3)^T$, and $a_1a_2a_3 = 0$. The condition $a_1a_2a_3 = 0$ implies that at least one of the elements of $\boldsymbol{a}$ is zero. Assuming $\boldsymbol{n} = (n_1,n_2,n_3)^T$, multiplying a_i to the ith scalar equation of (A.21) gives

$$\begin{aligned}-n_3a_1(\tilde{\boldsymbol{n}}_1)^T\tilde{F}\tilde{\boldsymbol{n}}_2 + n_2a_1(\tilde{\boldsymbol{n}}_1)^T\tilde{G}\tilde{\boldsymbol{n}}_2 &= 0\\ n_3a_2(\tilde{\boldsymbol{n}}_1)^T\tilde{E}\tilde{\boldsymbol{n}}_2 + n_1a_2(\tilde{\boldsymbol{n}}_1)^T\tilde{G}\tilde{\boldsymbol{n}}_2 &= 0\\ -n_2a_3(\tilde{\boldsymbol{n}}_1)^T\tilde{E}\tilde{\boldsymbol{n}}_2 + n_1a_3(\tilde{\boldsymbol{n}}_1)^T\tilde{F}\tilde{\boldsymbol{n}}_2 &= 0. \qquad \text{(A.22)}\end{aligned}$$

The above equation holds for every line. Now, we append one more subscript i to denote the corresponding values for the ith line. $rank(D) < 26$ if and only if the 26 rows of D are linearly independent. Considering the coefficients of elements of $\tilde{E}, \tilde{F}, \tilde{G}$ in (A.22), $rank(D) < 26$ if and only if

$$\sum_{i=1}^{13} (n_{3i}a_{2i} - n_{2i}a_{3i})\tilde{\boldsymbol{n}}_{li}\tilde{\boldsymbol{n}}_{2i} = \mathbf{0}$$
$$\sum_{i=1}^{13} (n_{1i}a_{3i} - n_{3i}a_{1i})\tilde{\boldsymbol{n}}_{li}\tilde{\boldsymbol{n}}_{2i} = \mathbf{0}$$
$$\sum_{i=1}^{13} (n_{2i}a_{li} - n_{li}a_{2i})\tilde{\boldsymbol{n}}_{1i}\tilde{\boldsymbol{n}}_{2i} = \mathbf{0} \qquad \text{(A.23)}$$

where $a_{1i}a_{2i}a_{3i} = 0$ to make sure at most two equations of (A.21) are used. Equation (A.23) can be rewritten using tensor notation

$$\sum_{i=1}^{13} (\boldsymbol{a}_i \times \boldsymbol{n}_i)\boldsymbol{n}_{1i}\boldsymbol{n}_{2i} = \mathbf{0}. \qquad \text{(A.24)}$$

Since $\boldsymbol{n}_i = \boldsymbol{x}_{pi} \times \boldsymbol{l}_i$, $i = 1, 2, \cdots, 13$, we will prove that for any projection normal $\boldsymbol{n}$, the following holds

$$\begin{aligned} &\left\{\boldsymbol{a} \times \boldsymbol{n} | \boldsymbol{a} = (a_1, a_2, a_3)^T, a_1a_2a_3 = 0\right\} \\ &= \{\boldsymbol{v} | \boldsymbol{v} \cdot \boldsymbol{n} = 0\} \\ &= \{a\boldsymbol{x}_p + b\boldsymbol{l} | \boldsymbol{n} = \boldsymbol{x}_p \times \boldsymbol{l},\ \text{any } a, b\} \end{aligned} \qquad \text{(A.25)}$$

where $\boldsymbol{x}_p$ is the position vector of the point on the line that is the closest to the origin, and $\boldsymbol{l}$ is the direction of the line. Then, (3.6) is equivalent to (A.24), which is equivalent to (3.4). Therefore, (3.4) has only trivial solutions if and only if (3.6) holds.

What remains to be proved is (A.25). The last equation is trivial since $\boldsymbol{n}$, $\boldsymbol{x}_p$ and $\boldsymbol{l}$ are orthogonal to one another. It is clear that $\{\boldsymbol{a} \times \boldsymbol{n} | \boldsymbol{a} = (a_1, a_2, a_3)^T, a_1a_2a_3 = 0\}$ is a subset of $\{\boldsymbol{v} | \boldsymbol{v} \cdot \boldsymbol{n} = 0\}$. Now, we prove the reverse. Let $\boldsymbol{s}$ be orthogonal to $\boldsymbol{n}$. Define $\boldsymbol{b} = \boldsymbol{s} \times \boldsymbol{n}$. Obviously, $\boldsymbol{b} \times \boldsymbol{n}$ is aligned with s. If $\boldsymbol{b}$ has a zero component, $\boldsymbol{a} = k\boldsymbol{b}$ gives $\boldsymbol{a} \times \boldsymbol{n} = \boldsymbol{s}$ for some real number k. Otherwise, $\boldsymbol{b} = (b_1, b_2, b_3)^T$ has no zero components. Noticing that $\boldsymbol{n}$ is not a zero vector, without loss of generality, assume its jth element is not zero $n_j \neq 0$. Then, let $\boldsymbol{a} = n_j\boldsymbol{b} - b_j\boldsymbol{n}$. $\boldsymbol{a}$ is a vector whose jth component is zero, and $\boldsymbol{a}$ is not aligned with $\boldsymbol{n}$ (since $b_j \neq 0$ and $n_j \neq 0$). $\boldsymbol{a}$ is a linear combination of $\boldsymbol{b}$ and $\boldsymbol{n}$; therefore, $\boldsymbol{a}$ is orthogonal to s. $k\boldsymbol{a} \times \boldsymbol{n} = \boldsymbol{s}$ for some real number k. Therefore, $\boldsymbol{s}$ is in the subset of the left-hand side of (A.25).□

ACKNOWLEDGMENT

The valuable comments of E. Altman are gratefully acknowledged.

REFERENCES

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

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

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

[4] J. Burns, A. Hanson, and E. Riseman, "Extracting straight lines," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-8, no. 4, pp. 425–455, July 1986.

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

[6] L. Davis, Z. Wu, and H. Sun, "Contour based motion estimation", *Comput. Vision Graphics Image Processing,* vol. 23, pp. 313–326, 1983.

[7] 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.

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

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

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

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

[12] Y. Liu and T. S. Huang, "Estimation of rigid body motion using straight line correspondences, further results," in *Proc. Int. Conf. Patt. Recog.* (Paris, France), Oct. 27–31, 1986, pp. 306–307.

[13] ——, "A linear algorithm for determining motion and structure from line correspondences, *Comput. Vision Graphics Image Processing,* vol. 44, no. 1, pp. 35–57, 1988.

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

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

[16] J. H. McIntosh and K. M. Mutch, "Matching straight lines," *Comput. Vision Graphics Image Processing,* vol. 43, no. 3, pp. 386–408, 1988.

[17] A. Mitiche, S. Seida, and J. K. Aggarwal, "Interpretation of structure and motion using straight line correspondences," in *Proc. Int. Conf. Patt. Recog.* (Paris, France), Oct. 27–31, 1986, pp. 1110–1112.

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

[19] M. D. Shuster, "Approximate algorithms for fast optimal attitude computation," *Proc. AIAA Guidance Cont. Spec. Conf.,* (Palo Alto), Aug. 1978, pp. 88–95.

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

[21] M. Spetsakis and J. Aloimonos, "Closed form solution to the structure from motion problem from line correspondences," in *Proc. Sixth AAAI Nat. Conf. Artificial Intell.* (Seattle, WA), July 1987, pp. 738–743.

[22] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of 3-D motion parameters of rigid bodies with curved surfaces, *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-6, no., 1, pp. 13–27, 1984.

[23] A. M. Waxman and K. Wohn, "Contour evolution, neighborhood deformation, and global image flow: Planar surfaces in motion," *Int. J. Robotics Res.* vol. 4, pp. 95–108, 1985.

[24] A. M. Waxman, B. Kamgar-Parsi, and M. Subbarao, "Closed-form solutions to image flow equations for 3D structure and motion," *Int. J. Comput. Vision,* vol. 1, pp. 239–258, 1987.

[25] J. Weng, N. Ahuja, and T. S. Huang, "Error analysis of motion parameter estimation from image sequences," in *Proc. First Int. Conf. Comput. Vision* (London, England), June 1987, pp. 703–707.

[26] ——, "Closed-form solution + maximum likelihood: A robust approach to motion and structure estimation," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.* (Ann Arbor, MI), June 5–9, 1988, pp. 381–386.

[27] J. Weng, T. S. Huang, and N. Ahuja, "Motion from images: Image matching, parameter estimation, and intrinsic stability," in *Proc. IEEE Workshop Visual Motion,* (Irvine, CA), Mar. 20–22, 1989, pp. 359–366

[28] ——, "Motion and structure from two perspective views: Algorithm, error analysis, and error estimation," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. 11, no. 5, pp. 451–476, 1989.

[29] B. L. Yen and T. S. Huang, "Determining 3-D motion and structure of a rigid body using straight line correspondences," *Image Sequence Processing and Dynamic Scene Analysis.* Heidelberg, Germany: Springer-Verlag, 1983.

[30] X. Zhuang and R. M. Haralick, "Rigid body motion and the optical flow image," in *Proc. 1st Conf. Artificial Intell. Applications,* (Denver, CO), Dec. 1984, pp. 366–375.

[31] X. Zhuang, T. S. Huang, and N. Ahuja, "A simplified linear optical flow-motion algorithm," *Compt. Vision Graphics, Image Processing,* vol. 42, pp. 334–344, 1988.

Juyang Weng (S'85–M'89) received the B.S. degree in computer science from Fudan University, Shanghai, China, in 1982. He received the M.S. and Ph.D. degrees both in computer science, from University of Illinois, Urbana-Champaign, in 1985 and 1988, respectively.

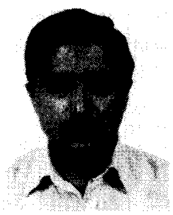
From September 1984 to December 1988, he was a research assistant at the Coordinated Science laboratory, University of Illinois, Urbana-Champaign. In the summer of 1987, he was employed at IBM Los Angeles Scientific Center, Los Angeles, CA. Since January 1989, he has been a researcher at Centre de Recherche Informatique de Montreal, Montreal, Canada, and an adjunct researcher at Ecole Polytechnique, University of Montreal, Montreal, Canada. Since October 1990, he has held a visiting assistant professor position at the University of Illinois, Urbana-Champaign. His current research interests include computer vision, image processing, object modeling and representation, parallel architecture for image processing, autonomous navigation, and artificial intelligence.

Thomas S. Huang (F'89) received the B.S. degree in electrical engineering from National Taiwan University, Taipei, Taiwan, and the M.S. and Sc.D. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, (MIT).

He was on the Faculty of the Department of Electrical Engineering at MIT 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 sabbatical leaves, he has worked at the MIT Lincoln Laboratory, the IBM Thomas J. Watson Research Center, and the Rheinishes Landes Museum in Bonn, West Germany, and has held visiting Professor positions at the Swiss Institutes of Technology in Zürich and Lausanne, University of Hannover in West Germany, and INRS-Telecommunications of the University of Quebec in Montreal, Canada. He has served as a consultant to numerous industrial firms and government agencies both in the United States and abroad. His professional interests lie in the broad area of information technology, especially the transmission and processing of multidimensional signals.

Dr. Huang has published 10 books and over 200 papers in network theory, digital filtering, image processing, and computer vision. His is a Fellow of the Optical Society of America, and has received a Guggenheim Fellowship (1971–72), an A. V. Humboldt Foundation Senior U.S. Scientist Award (1976–77), and a fellowship from the Japan Association for the Promotion of Science (1986). He received the IEEE Acoustics, Speech, and Signal Processing Society's Technical Achievement Award in 1987. 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 the 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 a 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. Since 1979, he has been with the University of Illinois at Urbana-Champaign, where since 1988, he has been a Professor in the Department of Electrical and Computer Engineering, the Coordinated Science Laboratory, and the Beckman Institute. His interests are in computer vision, robotics, image processing, and parallel algorithms. He has been involved in teaching, research, consulting and organizing conferences in these areas. His current research emphasizes integrated use of multiple image sources of scene information to construct 3D descriptions of scenes, the use of the acquired 3D information for object manipulations and navigation, and multiprocessor architectures for computer vision.

Dr. Ahuja was selected as a Beckman Associate in the University of Illinois Center for Advanced Study for 1990-91. He received the University Scholar Award (1985), Presidential Young Investigator Award (1984), National Scholarship (1967–72), and President's Merit Award (1966). He has coauthored the book *Pattern Models* (Wiley, 1983) with Schachter. He is Associate Editor of the journals IEEE Transactions on Pattern Analysis and Machine Intelligence, *Computer Vision, Graphics, and Image Processing,* and *Journal of Mathematical Imaging and Vision.* He is a member of the American Association for Artificial Intelligence, the Society of Photo-Optical Instrumentation Engineers, and the Association for Computing Machinery.