



0031-3203(94)00059-X

RIGID MOTION AND STRUCTURE FROM SEVERAL SETS OF PARALLEL LINES IN A MONOCULAR IMAGE SEQUENCE

SOO-CHANG PEI† and LIN-GWO LIOU

Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China

(Received 7 September 1993; in revised form 12 April 1994; received for publication 7 May 1994)

Abstract—An image line is a very reliable feature in the feature-based structure-from-motion (SFM) problem. It has at least three advantages: (1) it is not easily affected by occlusion. (2) It can achieve subpixel accuracy. (3) The correspondence problem is easier and less ambiguous. Although some generalized SFM algorithms using image line features have been recently developed, they usually suffer from high error sensitivity and large matrix size. Besides, in order to make their estimations more robust, many line correspondences (much more than the least requirement in theory) are usually needed. These shortages are not beneficial for practical use.

Because many sets of parallel lines are frequently encountered in an artificial 3D environment and can be easily identified, we try to utilize these identified parallel lines to simplify the generalized SFM problem and improve the performance of parameter estimation. A fast and reliable algorithm is proposed in our paper. Both computer-generated and real-world images are tested and several satisfactory results have been obtained. In addition, its error sensitivity and data degeneracy will be discussed.

Parallel lines Quaternion Perspective projection

1. INTRODUCTION

In the so-called structure-from-motion (SFM) problem, many kinds of image features such as points, line segments, lines and contours have been widely adopted for deriving the SFM algorithms. Here, line segments contain the information of their end-points but lines do not. An image line is considered as a fairly reliable feature for several reasons: (1) a line feature is not easily affected by occlusion: even if some part of an image line is missed or occluded, the whole feature may be completely reconstructed from its remaining parts; (2) position measurements of line features can achieve subpixel accuracy; (3) the problem for solving line correspondences is easier than that for point correspondences because more information (line orientation, length of line segment) can be utilized. Some related research about tracking image line segments (or lines) are listed in references (1–4). Other related research about motion estimation from image line or line segments are listed in references (5–15).

However, it is not easy to solve the SFM problem if prior information or some special motion and structure constraints are not available. It is usually because of the nonlinearity of their key equations. Wong *et al.*⁽⁵⁾ Liu and Huang⁽⁷⁾ Spetsatis and Aloimonos⁽⁶⁾ independently proposed similar generalized linear algorithms using at least 13 line correspondences at three views. Although a closed-form solution can be obtained, it

has several disadvantages: (1) its error sensitivity is high; (2) in order to solve the problem linearly, a lot of intermediate variables are defined and some important rigidity constraints are ignored. That is, many extra unknown variables need to be solved and the size of the linear system must be large; (3) in order to achieve a reliable estimate in practical application, the number of line correspondences is usually much larger than 13. If there are not enough line correspondences, a non-linear iteration method, such as (7, 9) should be adopted.

In fact, the generalized SFM algorithms^(5,6) are not very suitable for use in an artificial environment because some additional constraints about structure or motion are usually available. For example, there are some familiar constraints such as the ground planar constraint (GPC),⁽¹⁶⁾ the Legoland constraint⁽¹⁷⁾ (there exist three sets of parallel lines which are orthogonal to each other), and the epipolar constraint (used in a stereo vision system). Chen⁽¹⁰⁾ solves the motion problem by using line-to-plane correspondences. Sarali and Jong⁽¹¹⁾ and Zhang and Faugeras⁽¹²⁾ use line segments. Faugeras *et al.*⁽¹³⁾ discusses differential motion estimation by using line correspondences. Henriksen and Arnsparng⁽¹⁴⁾ propose a method for solving the orientation of a translating straight line in 3D space. These special SFM algorithms are usually simpler and much more reliable than the generalized ones.

Recently, set of parallel lines which are frequently encountered in an artificial environment have drawn a lot of researchers' attention. They are usually utilized

†Author to whom correspondence should be addressed.

in the applications of camera calibration.⁽¹⁷⁻²³⁾ Several methods about detecting parallel lines or vanishing points in 3D space were also proposed.⁽²⁴⁻²⁶⁾ In this paper, we extend the application of parallel lines to the area of motion estimation. After introducing this constraint, we hope the SFM problem can be solved more reliably and easily. In our algorithm, at least two sets of parallel lines in 3D space are required to solve the motion parameters. Because such a requirement can be very easily satisfied in a man-made environment, it is highly valuable in real applications. Experiments show a fairly good performance in estimating the desired motion and structure parameters. Both the computer-generated and real-world images are tested.

Our paper is organized as follows: Section 2, problem formulation; Section 3, description of the algorithm; Section 4, discussion about the algorithm; Section 5, experiments of computer simulations and real-world images; Section 6, conclusion; and finally, the Appendix.

2. PROBLEM FORMULATION

By examining Fig. 1, it can be seen that a rigid object moves in 3D space which is observed by a steady camera that obeys the rule of perspective projection. The viewing field of the camera is small (explained later). N_f frames are recorded as a monocular image sequence. In this image sequence, N image lines projected by the line features on the rigid object can be continuously traced. We also assume that the correspondence problem has been solved previously. For the N line features on the rigid object, some *unknown* part of them belong to several sets of parallel lines in 3D space. Now, our problem is: "How to solve the motion and structure of the rigid object by using the given N image line correspondences?".

3. ALGORITHM

Before introducing our algorithm, some variables are defined as follows for a more convenient description (also see Fig. 1).

The j th line feature on the rigid object at time t_i is denoted by L_{ij} . The orientation denoted by l_{ij} . The normal vector of the plane which passes through the focal center F and the image line l_{ij} is denoted by ϵ_{ij} ($\|\epsilon_{ij}\| = 1$). Here $i = 0$ to $N_f - 1$, $j = 1$ to N .

Because some unknown part of the line features on the rigid object belong to several sets of parallel lines in 3D space, the N indices of j can be considered as a union of several sets.

$$W = H \cup S_1 \cup S_2 \cup \dots \cup S_M. \tag{1}$$

Here, W is the universal index set $\{1, 2, \dots, N\}$, S_k is the set of indices which are corresponding to the k th set of parallel lines on the rigid object. There are totally M sets of parallel lines on the object. The set H contains other indices which do not belong to any one set of parallel lines. Because every line belongs to the index set S_k has the same orientation vector d_{ij} (at a fixed time t_i), we may define a new vector m_{ik} .

$$m_{ik} \equiv d_{ij}; \text{ where } j \in S_k. \tag{2}$$

The coordinate transform between a point P_0 on the rigid object at time t_0 and its corresponding point P_i at time t_i can be defined as

$$P_i = R_i P_0 + T_i, \tag{3}$$

where the matrix R_i is a 3×3 orthogonal matrix, and T_i is a 3×1 translation vector (see Fig. 2).

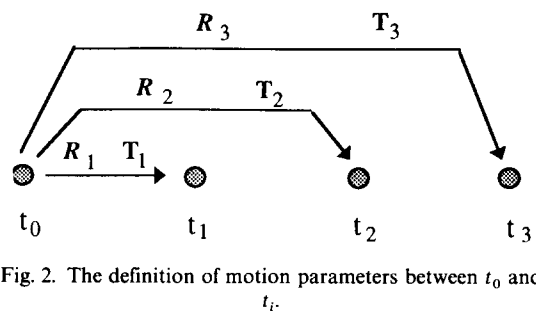


Fig. 2. The definition of motion parameters between t_0 and t_i .

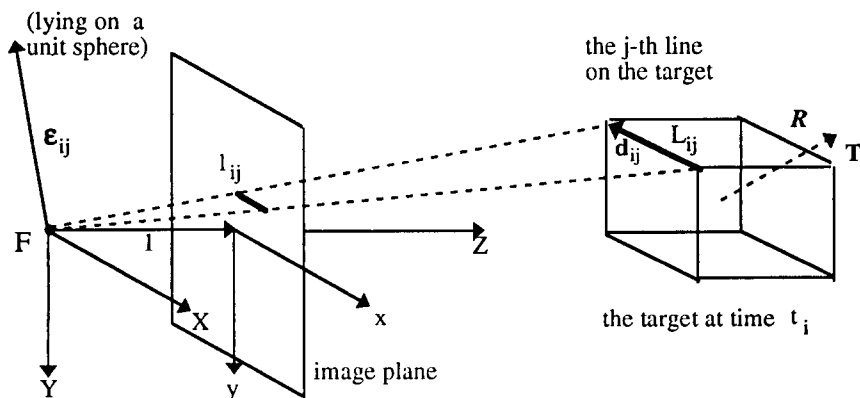


Fig. 1. Projection of the line features on the target. Here ϵ_{ij} is the normal vector of the plane passing through the focal point F and projected line l_{ij} .

Our algorithm is divided into three parts, described in the following three subsections.

3.1. Identifying the sets of parallel lines in 3D space

In the problem formulation, N image line features can be obtained at each one of the N_f frames. We also assume that some unknown portion is projected by several sets of parallel lines in 3D space (the image lines whose indices are in the index sets S_k s). A method for identifying the sets of parallel lines is proposed here. In the other words, we try to determine the elements in the index sets S_k s.

For providing more concise concepts to readers, a standard cube shown in Fig. 3(a) (the 0th image frame) is utilized to illustrate several important characters of our method. Figure 3(b) shows their corresponding vectors ϵ_{0j} s on a unit hemisphere. Note the directions of coordinate axes drawn on this figure. The circular band on this hemisphere represents the possible region where the vectors ϵ_{0j} s may exist (due to a small viewing field, see Appendix for more detail).

Before formally introducing our method, the concepts used in previous clustering methods⁽²⁴⁻²⁶⁾ are briefly discussed. Consider frame No. 0 (the frame grabbed at time t_0). For the image line l_{0j} whose index j belongs to the k th index set S_k , we have

$$\epsilon_{0j} \cdot \mathbf{m}_{0k} = 0, \quad j \in S_k. \tag{4}$$

This means that the cross-product (\times) of any two unit vectors ϵ_{0j_1} and ϵ_{0j_2} (here $j_1, j_2 \in S_k$) will be parallel to the direction vector \mathbf{m}_{0k} . We may define this unit vector as

$$\mathbf{n}_{j_1 j_2}^{(0)} \equiv \text{unit vector of } \{\epsilon_{0j_1} \times \epsilon_{0j_2}\}, \tag{5}$$

where the superscript (0) means the 0th image frame.

Therefore, previous clustering methods usually create an accumulation map on a unit hemisphere $Z < 0$ [see Fig. 3(c); please note the directions of the coordinate axes]. These methods pick every distinct pair of the N unit vectors ϵ_{0j} s (say $\{\epsilon_{0j_1}, \epsilon_{0j_2} | j_1, j_2 \in W\}$), calculate the unit vector $\mathbf{n}_{j_1 j_2}^{(0)}$, and add a constant value (say 1) to its corresponding position on the accumulation hemisphere. Besides, the unit vector $\mathbf{n}_{j_1 j_2}^{(0)}$ can also be explained as the unit vector of $[x, y, 1]^T$, where $[x, y]^T$ is the intersecting point of the two image lines l_{0j_1} and l_{0j_2} .

Theoretically, M highly accumulated peaks corresponding to \mathbf{m}_{0k} s can be found on this hemisphere [$M = 3$ in Fig. 3(c), represented by a mark “o”]. Except for these high peaks, there are many widely spread lower peaks created by the cross-products of ϵ_{0j} s from different index sets S_k s (represented by a mark “+”). It is obvious that the detection of high peaks will be considerably affected by these low peaks if observation errors exist. However, especially when the numbers of elements in the index sets (S_k s) are small, the performance of the accumulation hemisphere is very poor. See Fig. 3(a). It has 12 line features, and each one of the three parallel line sets has four lines. So we have a

total of $C_2^{12} = 66$ points on the accumulation hemisphere, and $3 \times C_2^4 = 18$ points of them will correspond to the desired three high peaks (height = 6). The other $66 - 18 = 48$ points, quite a large fraction of the total 66 points, are considered as *clutters*. The so-called clutters defined here are the clusters that do not contribute to finding the desired peaks. Unfortunately, these clutters are not usually widely spread and low enough. They deteriorate the detection of the correct high peaks.

From the discussion in Appendix, we know the projected image lines corresponding to the set S_k will look approximately parallel in a small viewing field if the angle θ_{0k} is large enough. This angle is defined as

$$\theta_{0k} \equiv \cos^{-1} (|\mathbf{m}_{0k} \cdot [0, 0, 1]^T|). \tag{6}$$

It also means that the position of their vanishing point on the image plane is far from the image origin $[0, 0]^T$. In other words, a parallel line set having a very small angle θ_{0k} will apparently look different and their vanishing point is near the image origin. Obviously, the region near the equator of the accumulation hemisphere corresponds to a large value of θ_{0k} , and the pole corresponds to a small value of θ_{0k} [see Fig. 3(c)]. For example, in Fig. 3(a), the image lines corresponding to the same parallel line set approximately look parallel and their corresponding clustering positions (marked by “o”) are all near the equator.

As to the wrong accumulating points (marked by “+”), we find that most of them are near the pole region. It is mainly because the intersecting point of two image lines l_{0j_1} and l_{0j_2} coming from different sets of parallel lines in 3D space is usually located within a small circular region around the image origin $[0, 0]^T$. See Fig. 3(c) again, as we can see that a large part of the wrongly accumulated positions in the pole region is corresponding to the nodes of the cube on the image plane. Notice the nodes of the cube connect a lot of non-parallel lines in 3D space.

After discussing the above concepts, it seems troublesome to detect all the parallel line sets by using the old clustering method. For example, it will be very hard to detect the set of parallel lines having a very small θ_{0k} . In our method to be proposed, we try to detect first the parallel line sets which look approximately parallel on the image plane. Then we delete these identified parallel lines from the universal index set W and test the remaining set. Our method is divided into the following steps:

Step 1. For every possible pair of the N image lines in frame 0, calculate their unit vectors $\mathbf{n}_{0j_1 j_2}$ [defined in equation (5)] if the following two constraints are satisfied: (1) the length of the image line segments corresponding to ϵ_{0j_1} and ϵ_{0j_2} must both be larger than a given threshold (say 30 pixels). It is because the vector ϵ of a short line segment is sensitive to image position errors. (2) A value ρ defined as $\|\epsilon_{0j_1} \times \epsilon_{0j_2}\|$ must be larger than a given threshold (say $\sin 5^\circ$). A larger value of ρ usually implies that the two image lines corresponding to ϵ_{0j_1} and ϵ_{0j_2} have a larger difference in

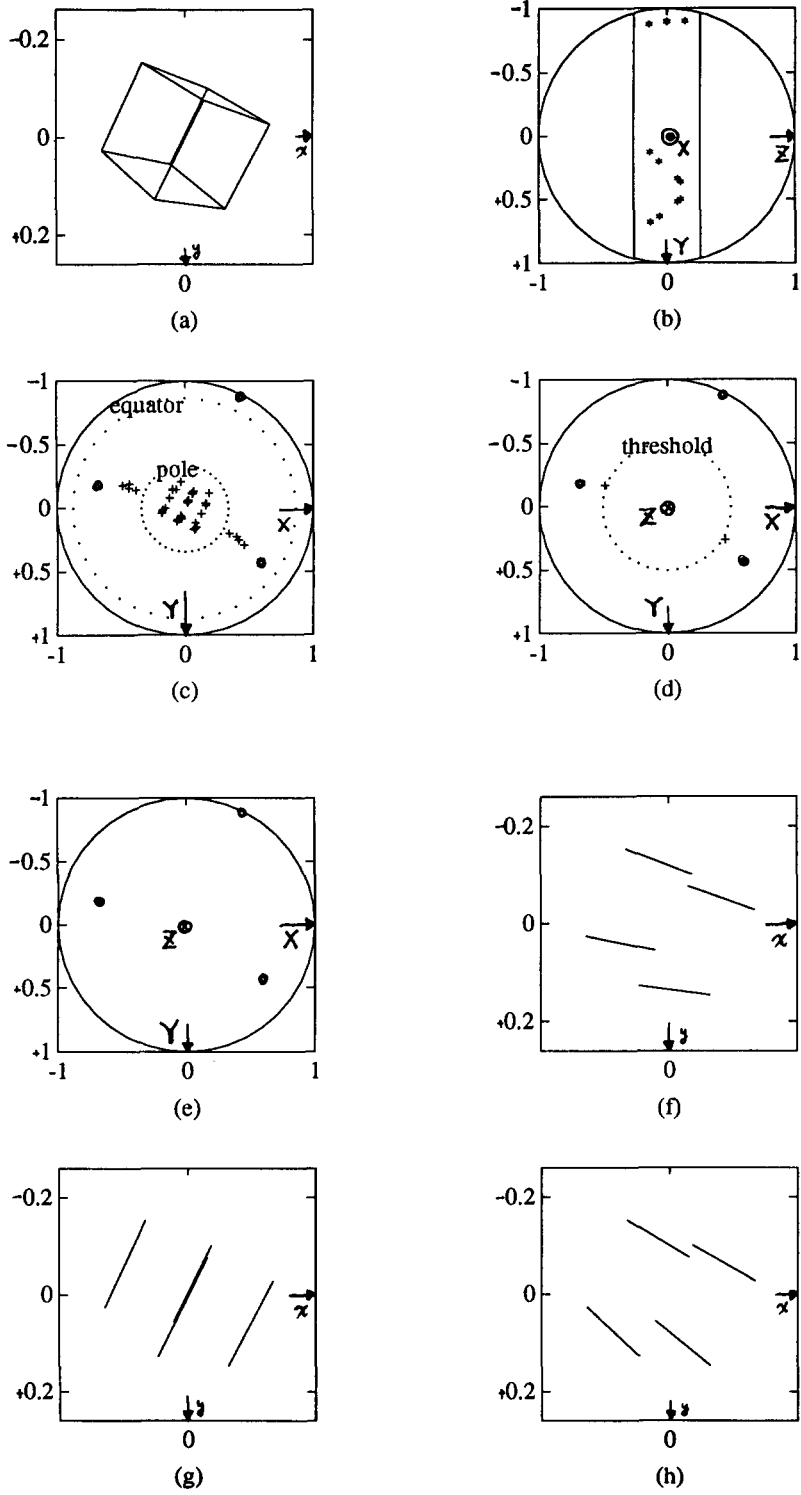


Fig. 3. An illustrating example for explaining the characteristics of the accumulation hemisphere.

their positions or directions. The estimate of the vector $\mathbf{n}_{j_1 j_2}^{(0)}$ will be poor if the positions of the two image lines are close together and almost parallel on the image plane.

Step 2. For the pairs passing through step 1, we can calculate an angle $\delta_{j_1 j_2}$ defined as

$$\delta_{j_1 j_2} \equiv \cos^{-1} (|\mathbf{n}_{j_1 j_2}^{(0)} \cdot [0, 0, 1]^T|), \quad (7)$$

where $\delta_{j_1 j_2}$ is in an interval $[0, \pi/2]$. If the calculated angle is higher than a given threshold θ_{th} (say 30°), add 1 to the corresponding position on the accumulation hemisphere. The pair not satisfying this constraint will be rejected. The final hemisphere will be similar to Fig. 3(d) – a very large part of the wrongly accumulating points near the pole are erased, and the performance of the clustering is highly improved. We call it the thresholded accumulation hemisphere.

Step 3. Erase the accumulating points which seem isolated on the thresholded accumulation hemisphere. This means that the points having no neighbors in their vicinity (a given suitable range) are erased from the hemisphere, as shown in Fig. 3(e). Therefore, the remaining highly accumulated positions can be easily identified and we will have a primary identification [see Fig. 3(f)–(h)]. Note that only frame No. 0 is utilized here.

Step 4. Because we have, in fact, N_f image frames, the results obtained by step 3 should be further tested in other frames to ensure the correctness of the identification. For the i th image frame, we may calculate the unit vectors $\mathbf{n}_{j_1 j_2}^{(i)}$ s whose indices j_1 and j_2 belong to a detected parallel line set from step 3. If the detected set is correct, all of the unit vectors $\mathbf{n}_{j_1 j_2}^{(i)}$ s will be parallel. If the detected set contains several wrong indices, the clustering analysis of the set at another frame should be performed and the wrong indices should be separated from the correct ones.

Step 5. If the number of the detected parallel line sets is larger than or equal to 2, the identification job is temporarily terminated because we do not need too many parallel line sets to estimate the motion and structure parameters (explained later). If the number

is less than 2, another frame can be chosen instead of the 0th frame.

Now, our identification method is completed.

3.2. Rotation estimation from the sets of parallel lines

After the sets of parallel lines are identified (at least two sets are needed), the motion parameters R_i and \mathbf{T}_i can be easily determined. First, we have

$$\mathbf{m}_{ik} = R_i \mathbf{m}_{0k}. \quad (8)$$

Therefore, a cost function $J(\mathbf{q}_i)$ can be defined as

$$J(\mathbf{q}_i) \equiv \sum_{k=1}^M |\mathbf{m}_{ik} - R_i \mathbf{m}_{0k}|^2, \quad (9)$$

where the vector $\mathbf{q}_i \equiv [\cos \theta_i/2, \sin \theta_i/2(n_{iX}, n_{iY}, n_{iZ})]^T$ is a 4×1 unit vector which is the quaternion expression of the rotation matrix R_i . The unit vector $\mathbf{n}_i \equiv (n_{iX}, n_{iY}, n_{iZ})^T$ is the rotation axis of R and θ_i is the rotation angle. Sometimes, we may also represent the rotation matrix by a 3D rotation vector defined as $\boldsymbol{\omega}_i \equiv \theta_i \mathbf{n}_i$.

Equation (9) can be rewritten in a new form

$$J(\mathbf{q}_i) = 2\mathbf{q}_i^T \mathbf{A}_i \mathbf{q}_i, \quad (10)$$

where

$$\mathbf{A}_i \equiv \sum_{k=1}^M \left(I - \begin{bmatrix} \mathbf{m}_{ik} \cdot \mathbf{m}_{0k} & -(\mathbf{m}_{ik} \times \mathbf{m}_{0k})^T \\ -(\mathbf{m}_{ik} \times \mathbf{m}_{0k}) & (\mathbf{m}_{ik} \mathbf{m}_{0k}^T + W_{ik} W_{0k}) \end{bmatrix} \right), \quad (11)$$

$$W_{ik} \equiv \begin{bmatrix} 0 & -\mathbf{m}_{ik,Z} & \mathbf{m}_{ik,Y} \\ \mathbf{m}_{ik,Z} & 0 & -\mathbf{m}_{ik,X} \\ -\mathbf{m}_{ik,Y} & \mathbf{m}_{ik,X} & 0 \end{bmatrix};$$

$$W_{0k} \equiv \begin{bmatrix} 0 & -\mathbf{m}_{0k,Z} & \mathbf{m}_{0k,Y} \\ \mathbf{m}_{0k,Z} & 0 & -\mathbf{m}_{0k,X} \\ -\mathbf{m}_{0k,Y} & \mathbf{m}_{0k,X} & 0 \end{bmatrix}. \quad (12)$$

Obviously, the quaternion vector \mathbf{q}_i that will minimize the cost function $J(\mathbf{q}_i)$ is just equal to the eigenvector corresponding to the smallest eigenvalue of the matrix \mathbf{A}_i . For a unique solution, the number M of the sets of parallel lines must be greater than or equal to 2, and the first component of \mathbf{q}_i [that is $\cos(\theta_i/2)$] must be positive. It is easy to determine the matrix R_i from \mathbf{q}_i by using the following relationship:

$$R = \begin{bmatrix} q_1^2 + q_2^2 - q_3^2 - q_4^2 & 2q_2q_3 - 2q_1q_4 & 2q_2q_4 + 2q_1q_3 \\ 2q_2q_3 + 2q_1q_4 & q_1^2 + q_3^2 - q_2^2 - q_4^2 & 2q_3q_4 - 2q_1q_2 \\ 2q_2q_4 - 2q_1q_3 & 2q_3q_4 + 2q_1q_2 & q_1^2 + q_4^2 - q_2^2 - q_3^2 \end{bmatrix}, \quad (13)$$

where $\mathbf{q} \equiv [q_1, q_2, q_3, q_4]^T$.

Because the rotation matrix can be solved by only two sets of parallel lines (at least), it does not matter whether we can identify all the M sets of parallel sets or not. Take the case of $M = 5$ for an example. If only the first two sets S_1 and S_2 are identified, the other three sets can be easily identified by solving their

corresponding orientation vectors \mathbf{d}_{0j} s ($j \in (W - S_1 - S_2)$). \mathbf{d}_{0j} s are determined by finding the eigenvector of the smallest eigenvalue of a matrix B_j defined as

$$B_j \equiv \sum_{i=0}^{N_f-1} (R_i^{-1} \boldsymbol{\varepsilon}_{ij})(R_i^{-1} \boldsymbol{\varepsilon}_{ij})^T \quad (14)$$

($R_0 = I$). Of no doubt, all of the lines having the same

orientation vector must belong to the same set of parallel lines. Then the rotation matrix R_i can be re-estimated by the total M sets of parallel lines, and the final estimation will be more accurate.

3.3. Structure and translation estimation

To specify the position of a line L_{0j} in 3D space, its orientation vector \mathbf{d}_{0j} and a point on this line must be given. Without loss of generality, we may define a point P_{0j} on the line L_{0j} , where $\overline{FP_{0j}} = k_{0j} \mathbf{h}_{0j}$. The three vectors $\{\boldsymbol{\varepsilon}_{0j}, \mathbf{d}_{0j}, \mathbf{h}_{0j}\}$ form an orthonormal basis in 3D space. Therefore, from equation (3), we have

$$[R_i(k_{0j} \mathbf{h}_{0j}) + \mathbf{T}_i] \cdot \boldsymbol{\varepsilon}_{ij} = a_{ij} k_{0j} + \boldsymbol{\varepsilon}_{ij} \cdot \mathbf{T}_i = 0, \quad (15)$$

where $a_{ij} \equiv R_i \mathbf{h}_{0j} \cdot \boldsymbol{\varepsilon}_{ij}$, $i = 1$ to $N_f - 1$, and $j = 1$ to N .

Because the size of the homogeneous linear system in equation (15) is large when N is larger, the computation load of directly solving equation (15) may be too heavy for us. So we make a slight modification to this system for reducing the computation load: To the j th line at N_f different time instants, we have

$$k_{0j} \mathbf{a}_j + E_j \mathbf{T} = 0, \quad (16)$$

where

$$\mathbf{a}_j \equiv [a_{1j}, a_{2j}, \dots, a_{N_f-1,j}]^T, \quad (17)$$

$$E_j \equiv \begin{bmatrix} \boldsymbol{\varepsilon}_{1j}^T & 0 & \dots & 0^T \\ 0^T & \boldsymbol{\varepsilon}_{2j}^T & \dots & 0^T \\ 0^T & 0^T & \dots & 0^T \\ 0^T & 0^T & \dots & \boldsymbol{\varepsilon}_{N_f-1,j}^T \end{bmatrix}, \quad (18)$$

and

$$\mathbf{T} \equiv [\mathbf{T}_1^T | \mathbf{T}_2^T | \dots | \mathbf{T}_{N_f-1}^T]^T. \quad (19)$$

Because \mathbf{a}_j is a $(N_f - 1) \times 1$ vector, there must exist another $N_f - 2$ unit vector $\mathbf{b}_{j,s}$ ($s = 1$ to $N_f - 2$) such that they are a set of orthogonal basis in $(N_f - 1) - D$ space. So we have

$$(|\mathbf{a}_j|) \mathbf{b}_{j,s}^T E_j \mathbf{T} \equiv \mathbf{u}_{j,s}^T \mathbf{T} = 0 \quad \text{for } s = 1 \text{ to } N_f - 2; j = 1 \text{ to } N, \quad (20)$$

where the additional multiplying scale $|\mathbf{a}_j|$ is a weighting factor which controls the contribution of equation (20). We may determine the vector \mathbf{T} (to a scale factor) by finding the eigenvector corresponding to the smallest

1. Loading the extracted feature lines of the given image sequence (correspondence problem is solved).
2. From the method described in section III.1 (five steps), we can obtain a primary identification of parallel lines in 3D space.
3. From the above identification, rotation parameters R_i s can be estimated by using eq.(9) and eq.(13).
4. From the estimated rotation, we can approximately determine the orientation vector \mathbf{d}_{0j} of every feature lines on the 3D object.
5. From the determined orientation vectors \mathbf{d}_{0j} s, we may detect the remaining parallel line sets (if they exist) which were not successfully detected by step 2.
6. If new sets of parallel lines are detected, rotation parameters R_i s can be re-estimated for a more accurate result. If no new set exists, go to step 7.
7. Estimate the translation vectors $\mathbf{T}_1, \mathbf{T}_2, \dots$ and depth values k_{0j} s by using eq.(20) and eq.(22).

Fig. 4. Summary of the whole algorithm.

eigenvector of a matrix C defined as

$$C \equiv \sum_{j=1}^N \sum_{s=1}^{N_f-2} \mathbf{u}_{js} \mathbf{u}_{js}^T. \quad (21)$$

Equation (20) is a homogeneous linear system with $3(N_f - 1)$ unknown variables and $N(N_f - 2)$ equations. The vector \mathbf{T} can be solved to a scale factor if $N_f \geq 3$ and $N \geq (3N_f - 4)/(N_f - 2)$. Then the value k_{0j} can be easily determined by using

$$k_{0j} = -\frac{\mathbf{a}_j^T E_j \mathbf{T}}{\mathbf{a}_j^T \mathbf{a}_j}. \quad (22)$$

The whole algorithm is completed here. Figure 4 shows an overview of it.

4. DISCUSSION

4.1. How many lines and frames are needed?

Theoretically, from equation (8), at least two sets of corresponding vectors $\{(\mathbf{m}_{0k}, \mathbf{m}_{ik}) | k = 1, 2, \dots, M, M \geq 2\}$ are needed to uniquely specify the rotation R_i between two time instants t_0 and t_i . From the known rotation, the orientation vectors \mathbf{d}_{0j} s of the lines on the 3D object can be easily reconstructed. That is to say, it is enough to solve the rotation and line orientations of the object by using only two frames and two parallel line sets. Besides, every parallel line set contains at least three lines (otherwise, the method for identifying the parallel lines would fail). Therefore, the smallest total number N of lines on the 3D object is 6.

However, from equations (15) and (16), the translation \mathbf{T} and the line depths k_{0j} s cannot be solved by using only two frames. It needs at least three frames ($N_f \geq 3$) and usually more than five line features on the object [in fact, $N \geq (3N_f - 4)/(N_f - 2)$].

From the above discussion, at least six image line correspondences at three frames are needed to solve the SFM problem completely. These six image line correspondences must belong to two sets of parallel lines in 3D space (three for each one).

4.2. Error sensitivity

In our algorithm, rotation matrices R_i s must be first solved. What factors will affect their estimations? From equation (10), we know the rotation R_i can be directly calculated by its corresponding quaternion \mathbf{q}_i , the eigenvector corresponding to the smallest eigenvalue of the matrix A_i . Because matrix A_i is constructed by the orientation vectors of parallel lines in 3D space (that is $\{(\mathbf{m}_{0k}, \mathbf{m}_{ik}) | k = 1, 2, \dots, M\}$), the relationship between these orientation vectors directly influence the error sensitivity of \mathbf{q}_i . For simplicity, if only two sets of parallel lines ($M = 2$) are considered, we have the following rules: (1) the estimation is least error-sensitive when the opening angle between \mathbf{m}_{01} and \mathbf{m}_{02} is 90° ; (2) the smaller the opening angle between \mathbf{m}_{01} and \mathbf{m}_{02} , the more error-sensitive the estimation becomes; (3) the estimation fails when $\mathbf{m}_{01} = \mathbf{m}_{02}$ (singular case). The above rules will be proved in the experiments.

While estimating the translation \mathbf{T} defined in equation (19), there exists a situation which will let the estimation fail. Take $N_f = 3$ for an example. From Section 3.3, we know at least three frames are needed for solving the translation $\mathbf{T}^T = [\mathbf{T}_1^T | \mathbf{T}_2^T]$ by using the homogeneous linear system described in equation (20). If $\mathbf{T}_1 = 0$ (only rotation between t_0 and t_1), is there any possibility that we can still obtain \mathbf{T}_2 by using equation (20)? The answer is NO! Because \mathbf{T}_1 is equal to zero, the constant a_{1j} s defined in equation (15) are all equal to zero for $j = 1$ to N . Although a_{2j} s are not zero, the translation \mathbf{T}_2 is still unsolvable. So it is a data degeneracy caused by no-translation. In fact, we may consider the image frame captured at t_1 as a *null* frame which has no contribution to the estimation of translation. This means that the effective number of the given image sequence is only 2, not enough for the estimation (at least three are needed). Therefore, we think that translation vector \mathbf{T} is solvable if the effective number of frames $N_{\text{eff}} (= N_f - N_{\text{null}})$ is greater than or equal to 3. Experiments support the above thoughts.

5. EXPERIMENTS

In this section, two kinds of image data are tested—computer-generated data and real-world image data. Several things need to be proved in the following experiments: (1) our algorithm is exactly accurate if no observation errors exist; (2) our algorithm can handle the real-world image data and obtain a satisfactory estimation; (3) our algorithm is robust and not easily affected by observation errors; (4) error-sensitivity depends on the structure of feature lines on the object; (5) data degeneracy happens when the effective number N_{eff} is less than 3.

Before introducing these experiments, we first specify the method of adding simulated observation errors on the image feature lines. In Section 3, we know an image line l_{ij} can be represented by a unit vector \mathbf{e}_{ij} . Assume $\mathbf{u}_{iJ}, \mathbf{v}_{iJ}$ and \mathbf{e}_{ij} form an orthonormal basis in 3D space. The error-perturbed version of \mathbf{e}_{ij} is defined as

$$\text{error-perturbed } \mathbf{e}_{ij} \equiv \text{unit vector of } \{\mathbf{e}_{ij} + \sigma_1 \mathbf{u}_{ij} + \sigma_2 \mathbf{v}_{ij}\}, \quad (23)$$

where σ_1 and σ_2 are random variables of uniform distribution $[-\sigma, \sigma]$. In later experiments concerning adding observation errors, the value of σ controls the magnitude of these errors. If the opening angle of the camera is 30° and the image plane's resolution is 512×512 , $\sigma = 0.001$ will correspond to a maximum position deviation of one pixel. Because σ is usually small, the σ value can be represented by rad or degree. For examples, $\sigma = 0.001$ is equivalent to $\sigma = 0.001$ rad or $\sigma = 0.0586^\circ$.

5.1. Computer simulation

In this subsection, a computer-generated rigid object is adopted. Here $N = 20$ and $N_f = 4$. Its motion parameters R_i and \mathbf{T}_i (for $i = 1-3$) are randomly assigned.

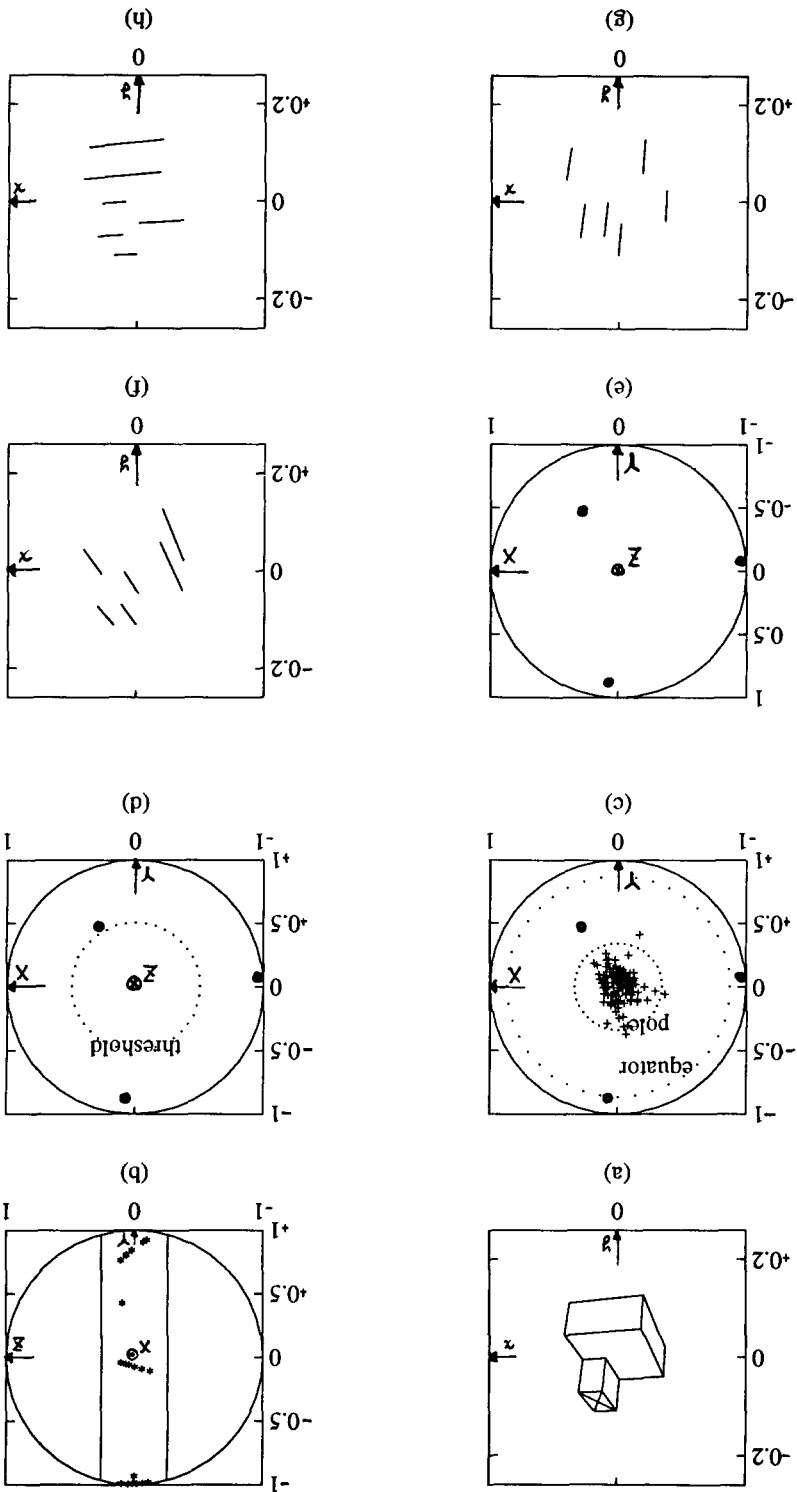


Fig. 5. The accumulation hemisphere and the final detected sets of parallel lines for the experiment described in Section 5.1.

Table 1. Error-free estimation ($\sigma = 0^\circ$) for the line structure shown in Fig. 5(a)

Parameters	Real	Estimate	Error(%)
Ω_1	(0.3000, -0.2000, 0.1000) ^T	(0.3000, -0.2000, 0.1000) ^T	$0.3798 \times 10^{-11}\%$
Ω_2	(0.6000, -0.5000, 0.2000) ^T	(0.6000, -0.5000, 0.2000) ^T	$0.5470 \times 10^{-11}\%$
Ω_3	(0.8000, -0.7000, 0.3000) ^T	(0.8000, -0.7000, 0.3000) ^T	$0.5317 \times 10^{-11}\%$
T_1	(3.6108, 6.0587, -0.7151) ^T	(3.6108, 6.0587, -0.7151) ^T	$0.1798 \times 10^{-10}\%$
T_2	(7.8149, 11.6886, 1.7767) ^T	(7.8149, 11.6886, 1.7767) ^T	$0.5712 \times 10^{-10}\%$
T_3	(9.1563, 14.8344, 5.1969) ^T	(9.1563, 14.8344, 5.1969) ^T	$0.6704 \times 10^{-10}\%$
\mathbf{m}_{01}	(0.2833, 0.4721, -0.8348) ^T	(0.2833, 0.4721, -0.8348) ^T	0.0854×10^{-5} deg
\mathbf{m}_{02}	(-0.0729, 0.8785, 0.4721) ^T	(-0.0729, 0.8785, 0.4721) ^T	0.1207×10^{-5} deg
\mathbf{m}_{03}	(0.9563, -0.0729, 0.2833) ^T	(0.9563, -0.0729, 0.2833) ^T	0.0854×10^{-5} deg

Table 2. Estimation of the line structure shown in Fig. 5(a) when observation errors exist ($\sigma = 0.25^\circ$)

Parameters	Real	Estimate	Error(%)
Ω_1	(0.3000, -0.2000, 0.1000) ^T	(0.3084, -0.1957, 0.0971) ^T	2.6441%
Ω_2	(0.6000, -0.5000, 0.2000) ^T	(0.6013, -0.5063, 0.2004) ^T	0.8033%
Ω_3	(0.8000, -0.7000, 0.3000) ^T	(0.8014, -0.7029, 0.2997) ^T	0.2913%
T_1	(3.6108, 6.0587, -0.7151) ^T	(3.5943, 6.1872, -0.7394) ^T	1.8601%
T_2	(7.8149, 11.6886, 1.7767) ^T	(7.8681, 11.7453, 1.5090) ^T	1.9666%
T_3	(9.1563, 14.8344, 5.1969) ^T	(9.1705, 14.8814, 4.7171) ^T	2.6510%
\mathbf{m}_{01}	(0.2833, 0.4721, -0.8348) ^T	(0.2803, 0.4663, -0.8391) ^T	0.4513 deg
\mathbf{m}_{02}	(-0.0729, 0.8785, 0.4721) ^T	(-0.0732, 0.8782, 0.4727) ^T	0.0400 deg
\mathbf{m}_{03}	(0.9563, -0.0729, 0.2833) ^T	(0.9583, -0.0727, 0.2764) ^T	0.4099 deg

The test object is shown in Fig. 5(a). Its accumulation behavior is shown in Fig. 5(b)–(e) which are similarly defined as in Fig. 3(b)–(e). The detected sets of parallel lines are shown in Fig. 5(f)–(h). Table 1 lists the estimation of motion and line orientations without adding any observation errors. The estimation is very accurate, which proves our algorithm. Table 2 lists the estimation, purposely adding observation errors ($\sigma = 0.25^\circ$). The results shown in Table 2 are still quite satisfactory. Notice that the estimation of translation vectors are appropriately scaled, and the estimated rotation matrices R_i s are represented by their corresponding rotation vectors ω_i for reducing the size of the tables.

5.2. Real-world image tests

In this subsection, a real-world image sequence (three frames) is tested. They are shown in Fig. 6(a)–(c). This image sequence is acquired by a moving camera which passes through a corridor (a large rigid object). This camera is translating to the Z direction and rotating about the Z-axis.

Because the image line extraction and tracking algorithm is not the main consideration in this paper, we only briefly describe our process of line extraction in the following steps: (1) the image is first smoothed to reduce the effect of noise; (2) apply an edge operator to this smoothed image and preserve the edges which are stronger than a threshold; (3) the edge points supporting a physical line segment are collected together and the best-fitted line equation is then determined by linear regression. Notice that the edges shorter than a

threshold will be discarded; (4) although there have been a lot of successful automatic line-tracking methods recently,^(3,8,9,23) the image line features showing up in every frame of the image sequence are manually selected and corresponded for simplifying our processing. It is because the final 3D motion estimation will not be affected by method we use to determine the correct correspondences; (5) from the results of camera calibration, the length unit of these line equations can be changed from one image pixel to one focal length for matching our mathematical derivations. The image line equations with new length units are then stored in computer as our input data. Figure 6(d) shows the final extracted line features in frame No. 0.

After applying the input image data to our algorithm, we first determine two sets of parallel lines [$\mathbf{m}_{01} = (0, 1, 0)^T$ and $\mathbf{m}_{02} = (-1, 0, 0)^T$, denoted by “o”] from the thresholded accumulation hemisphere shown in Fig. 6(e). We can find that a very large part of the wrongly accumulated points (denoted by “+”) are rejected by the threshold circle ($\theta_{th} = 30^\circ$). Although only two sets of parallel lines are identified, our algorithm still works and the third parallel line set [$\mathbf{m}_{03} = (0, 0, 1)^T$] can be recovered later. The final estimation results are listed in Table 3. The estimation errors for orientation vectors \mathbf{m}_{0k} s are very small (about 1°). Because the magnitudes of the rotation vectors ω_i s are not large, it is quite natural that the percentage estimation errors for ω_i s will become larger. For example, ω_1 corresponds to the rotating around the Z-axis by 8° , and ω_2 by 18° . The percentage errors of 15.6% and 7.7% separately correspond to the errors of only 1.25°

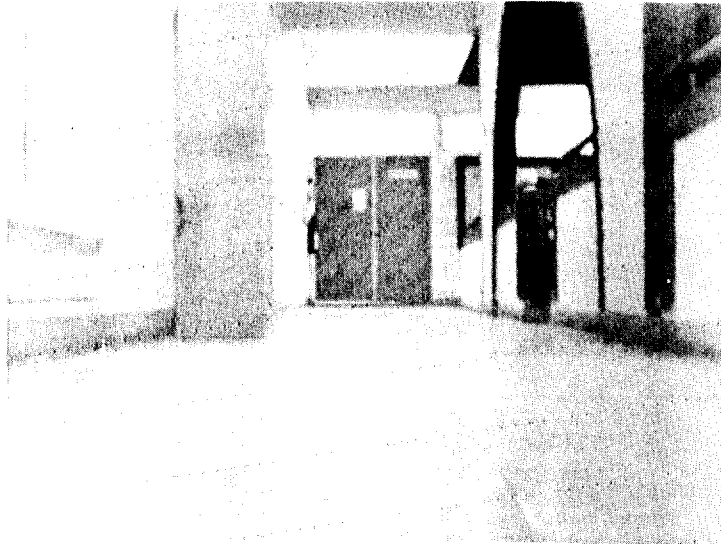


Fig. 6. (a) Frame No. 0.



Fig. 6. (b) Frame No. 1.



Fig. 6. (c) Frame No. 2.

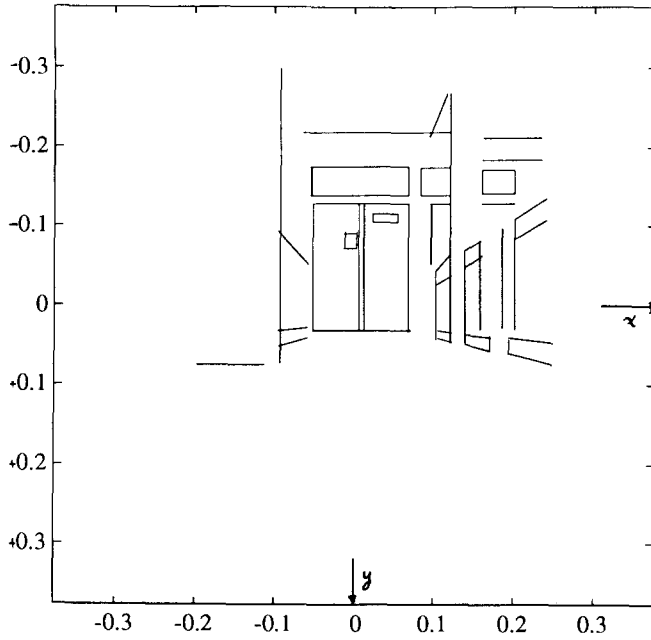


Fig. 6. (d) Extracted line features of frame No. 0.

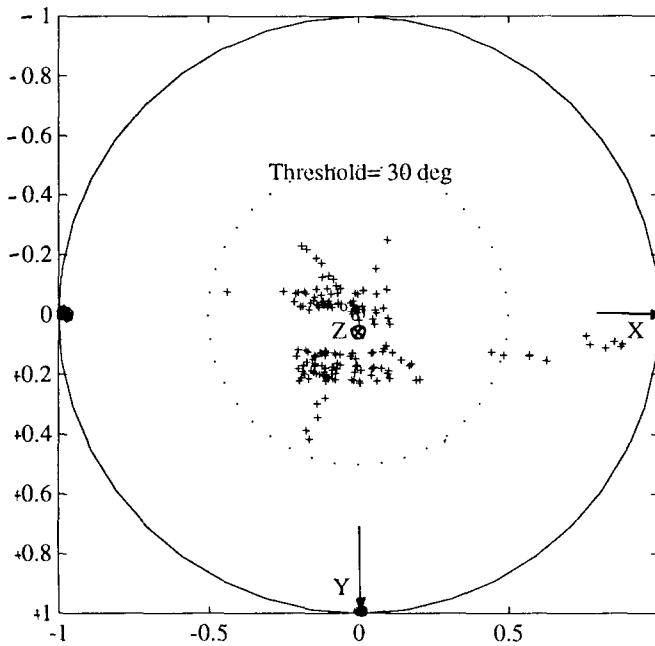


Fig. 6. (e) Thresholded accumulation hemisphere.

Fig. 6. The test for the real-world image sequence described in Section 5.2.

and 1.39° , respectively, which are in fact very small. Note the estimation for translation vectors T_1 and T_2 have been appropriately scaled.

Two kinds of errors may affect our results: (1) image position errors; (2) camera calibration errors. The influences from the calibration errors should be especially noticed. Besides, the provided 'true answers' listed on Table 3 are not precisely accurate because they are measured quantities. So, the estimation errors listed on Table 3 are just calculated for an approximation.

5.3. Error sensitivity and data degeneracy

In this subsection, we want to analyze the error sensitivity of our algorithm. Two experiments are executed.

The first experiment tests the error performance of a variable structure shown in Fig. 7. This structure has two sets of parallel lines which are lying on the same plane. Each set has five lines of equal length 2. Their intersecting angle and width are separately denoted by

Table 3. Estimation of the line structure shown in the real-world image sequence

Parameters	Measure	Estimate	Error(%)
Ω_1	$(0.0000, 0.0000, 0.1396)^T$	$(-0.0118, 0.0181, 0.1366)^T$	15.626%
Ω_2	$(0.0000, 0.0000, 0.3140)^T$	$(-0.0138, 0.0192, 0.3089)^T$	7.7034%
T_1	$(0.0000, 0.0000, -6.9204)^T$	$(-0.3782, -0.4184, -6.9854)^T$	8.2035%
T_2	$(0.0000, 0.0000, -13.8409)^T$	$(-0.7132, -0.5079, -13.7254)^T$	6.3807%
m_{01}	$(0.0000, 1.0000, 0.0000)^T$	$(0.0145, 0.9996, -0.0241)^T$	1.612 deg
m_{02}	$(1.0000, 0.0000, 0.0000)^T$	$(0.9997, 0.0147, -0.0179)^T$	1.327 deg
m_{03}	$(0.0000, 0.0000, 1.0000)^T$	$(0.0033, -0.0062, 1.0000)^T$	0.402 deg

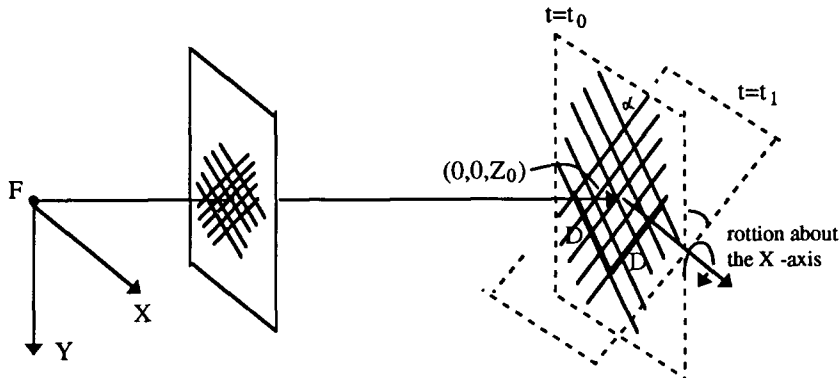


Fig. 7. The test structure used in the error analysis described in Section 5.3. Both of the two parallel line sets are lying on the same plane in 3D space. The center of the structure is kept at $(0, 0, Z_0)$ during the motion.

Table 4. Motion estimation of the two cases which have different effective frame numbers ($N_{eff} = 3$ and $N_{eff} = 2$)

Case	Parameters	Real	Estimate
Case 1	Ω_1	$(0.0000, 0.0000, 0.3000)$	$(0.0000, 0.0000, 0.3000)$
	Ω_2	$(0.0000, 0.0000, 0.6000)$	$(0.0000, 0.0000, 0.6000)$
	Ω_3	$(0.3000, 0.2000, 0.8000)$	$(0.3000, 0.2000, 0.8000)$
	T_1	$(0.0000, 0.0000, 0.0000)$	$(0.0000, 0.0000, 0.0000)$
	T_2	$(0.0000, 0.0000, 3.0000)$	$(0.0000, 0.0000, 3.0000)$
	T_3	$(-5.7560, 3.7592, 6.2187)$	$(-5.7560, 3.7592, 6.2187)$
Case 2	Ω_1	$(0.0000, 0.0000, 0.3000)$	$(0.0000, 0.0000, 0.3000)$
	Ω_2	$(0.0000, 0.0000, 0.6000)$	$(0.0000, 0.0000, 0.6000)$
	Ω_3	$(0.3000, 0.2000, 0.8000)$	$(0.3000, 0.2000, 0.8000)$
	T_1	$(0.0000, 0.0000, 0.0000)$	$(0.0000, 0.0000, 0.0000)$
	T_2	$(0.0000, 0.0000, 0.0000)$	$(0.0000, 0.0000, 0.0000)$
	T_3	$(-5.7560, 3.7592, 6.2187)$	(Not unique)

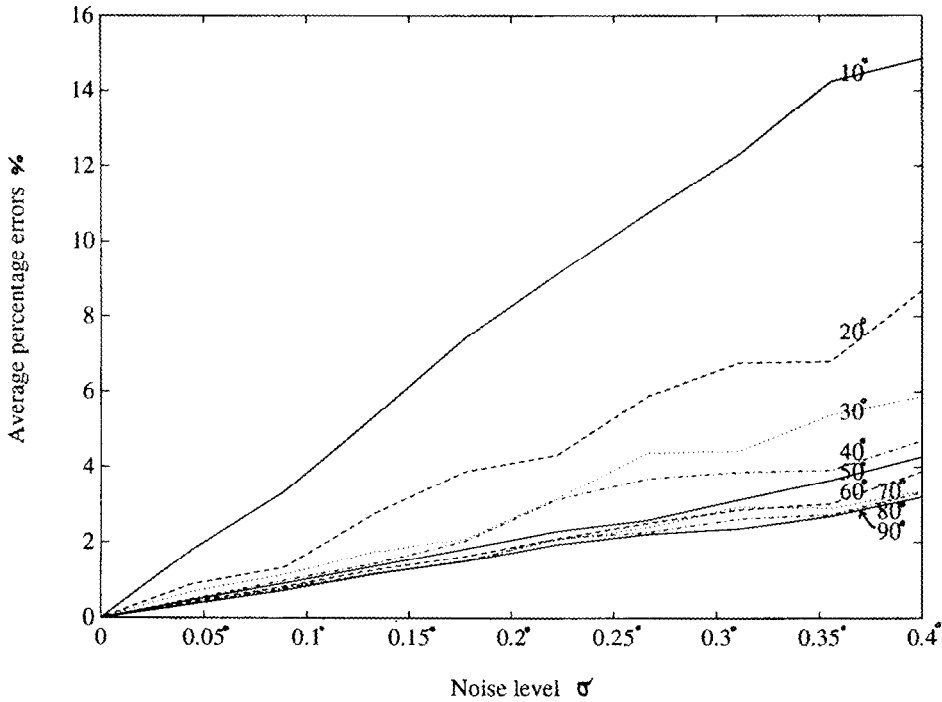


Fig. 8. (a) Error performance analysis of the test object shown in Fig. 7. Here, the width D is fixed to 1; however, the intersecting angle α varies from 10° to 90° . It is obvious that the structure corresponding to 90° has the best error performance.

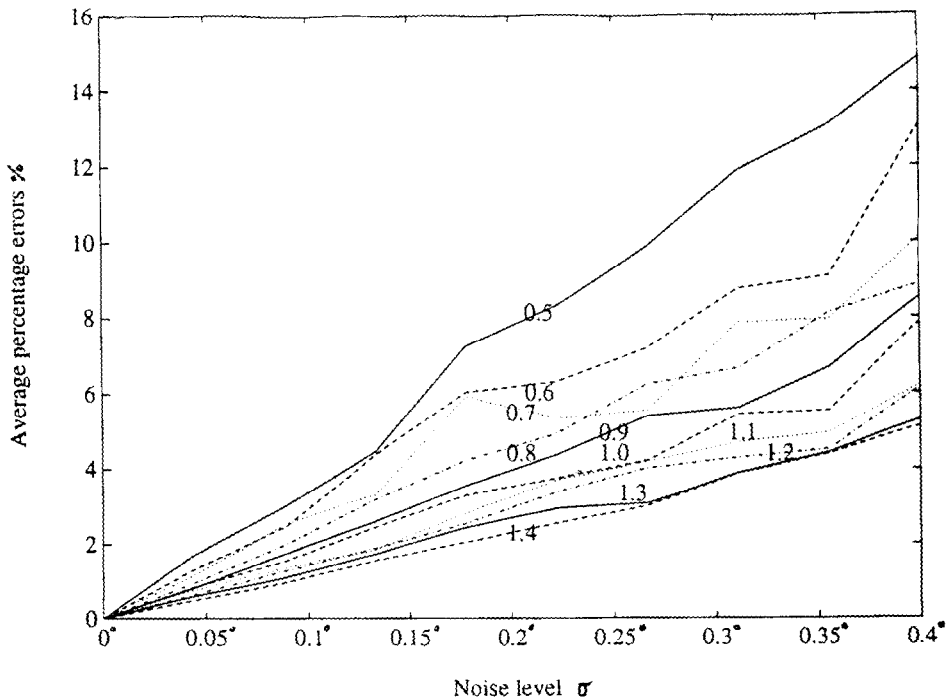


Fig. 8. (b) Error performance analysis of the test object shown in Fig. 7. Here, the intersecting angle α is fixed to 90° ; however, the width D varies from 0.5 to 1.4. It is obvious that the structure corresponding to $D = 1.4$ has the best error performance.

α and D . The center of the structure is fixed at $(0, 0, 8)^T$. At time t_0 , the normal vector of the plane is $(0, 0, 1)^T$; at time t_1 , we rotate this plane structure around the point $(0, 0, 8)^T$ by a rotation vector $(1, 0, 0)^T$. Only the percentage errors of rotation estimation are considered here.

Figure 8(a) shows the error performance of the cases when D is fixed to 1.0 and α is changing from 10° to 90° . It is easy to find that the error performance is best when $\alpha = 90^\circ$. Figure 8(b) shows the error performance of the cases when α is fixed to 90° and D is changing from 0.5 to 1.4. The case when $D = 1.4$ has the best performance. Every point on the curves shown in Fig. 8 are calculated by averaging 100 iterations.

The second experiment shows that the translation vector \mathbf{T} can be solved if and only if the effective frame number N_{eff} is greater than or equal to 3. The test object used here is the same as that used in Section 5.1. However, we purposely assign some of the translation \mathbf{T}_i to zero in order to change the effective frame number N_{eff} . See Table 4. The translation \mathbf{T}_1 is set to zero in the case 1 ($N_{\text{eff}} = 4 - 1 = 3$); however, both the translations \mathbf{T}_1 and \mathbf{T}_2 are set to zero in the case 2 ($N_{\text{eff}} = 4 - 2 = 2$). In both cases, their rotation parameters are successfully recovered. But, only the estimation of translation vectors in case 1 succeed. The matrix C [defined in equation (21)] in case 1 has only one zero eigenvalue, but the matrix C in case 2 has three zero eigenvalues. That is why case 2 fails to determine the translation uniquely. It proves the statement made in Section 4.2.

6. CONCLUSION

Although some algorithms for solving the generalized SFM problem by using line correspondences have been developed, they suffer from the high error-sensitivity and large number of required correspondence sets. In real applications, these shortages usually puzzle us. In this paper, the sets of parallel lines frequently encountered in an artificial environment are utilized for simplifying the generalized SFM problem. After identifying at least two sets of parallel lines, we can solve an initial guess of rotation parameters by utilizing the expression of quaternion. From this initial solution, the other unidentified sets of parallel lines can be determined and the solution of rotation parameter can be further modified. After that, the translation and structure of the object can be obtained. In our algorithm, the computational complexity is greatly reduced and the reliability of the estimation is highly improved. Simulation and real-world image experiments prove these results. Therefore, our algorithm seems quite valuable in real applications.

In future, we hope to develop some similar algorithms using other kinds of special constraints encountered in the daily environment for preventing the difficulty and high error-sensitivity when using a generalized SFM algorithm.

REFERENCES

1. Yuncai Liu and T. S. Huang, Determining straight line correspondences from intensity images, *Pattern Recognition* **24**, 489–504 (1991).
2. H. H. Chen and T. S. Huang, Matching 3-D line segments with applications to multiple-object motion estimation, *IEEE Trans Pattern Anal. Mach. Intell.* **12** (October 1990).
3. R. Deriche and O. Faugeras, Tracking line segments, *Image Vision Computing* **8** (November 1990).
4. J. Princen, J. Illingworth and J. Kittler, A hierarchical approach to line extraction based on the Hough transform, *Computer Vision Graphics Image Process.* **52**, 57–77 (1990).
5. Juyang Weng, T. S. Huang and N. Ahuja, Motion and structure from line correspondences: closed-form solution, uniqueness, and optimization, *IEEE Trans Pattern Anal. Mach. Intell.* **14** (March 1992).
6. M. E. Spetsakis and J. Aloimonos, Structure from motion using line correspondences, *Int. J. Computer Vision* **4**, 171–183 (1990).
7. Yuncai Liu and T. S. Huang, A linear algorithm for motion estimation using straight line correspondences, *Computer Vision Graphics Image Process.* **44**, 35–37 (1988).
8. Yuncai Liu and T. S. Huang, Three-dimensional motion determination from real scene images using straight line correspondences, *Pattern Recognition* **25**, 617–639 (1992).
9. Yuncai Liu, T. S. Huang and O. D. Faugeras, Determination of camera location from 2D to 3D line and point correspondences, *IEEE Trans Pattern Anal. Mach. Intell.* **12** (January 1990).
10. H. H. Chen, Pose determination from line-to-plane correspondences: existence condition and closed-form solutions, *IEEE Trans Pattern Anal. Mach. Intell.* **13** (June 1991).
11. E. Salari and Chiou-Muh Jong, A method to calculate the structure and motion parameters from line correspondences, *Pattern Recognition* **23**, 553–561 (1990).
12. Zhengyou Zhang and O. D. Faugeras, Determining motion from 3D line segment matches: a comparative study, *Image Vision Computing* **9** (February 1991).
13. O. D. Faugeras, N. Navab and R. Deriche, Information contained in the motion field of lines and the cooperation between motion and stereo, *Int. J. Imaging Systems Technol.* **2**, 356–370 (1990).
14. K. Henriksen and J. Arnspar, Direct determination of the orientation of a translating 3D straight line, *Pattern Recognition Lett.* **10**, 251–258 (1989).
15. Ming Xie, 3D Motion estimation from combined 2D–3D data of line segments, *Pattern Recognition Lett.* **14**, 191–197 (1993).
16. T. N. Tan, K. D. Baker and G. D. Sullivan, 3D Structure and motion estimation from 2D image sequences, *Image Vision Computing* **11** (May 1993).
17. M. Straforini, C. Coelho and M. Campani, Extraction of vanishing points from images of indoor and outdoor scenes, *Image Vision Computing* **11** (March 1993).
18. B. Caprile and V. Torre, Using vanishing points for camera calibration, *Int. J. Computer Vision* **4**, 127–140 (1990).
19. W. Chen and B. C. Jiang, 3D camera calibration using vanishing point concept, *Pattern Recognition*.
20. Zen Chen, Din-Chang Tseng and Jenn-Yee Lin, A simple vision algorithm for 3D position determination using a single calibration object, *Pattern Recognition* **22**, 173–187 (1989).
21. T. Echigo, A camera calibration technique using three sets of parallel lines, *Mach. Vision Applic.* **3**, 159–167 (1990).
22. Ling-Ling Wang and Wen-Hsiang Tsai, Computing camera parameters using vanishing-line information from

- a rectangular parallelepiped, *Machine Vision Applic.* 3, 129-141 (1990).
23. B. Brillault-O'Mahony, High level 3D structures from a single view, *Image Vision Computing* 10 (September 1992).
 24. Long Quan and R. Mohr, Determining perspective structures using hierarchical Hough transform, *Pattern Recognition Lett.* 9, 279-286 (1989).
 25. M. J. Magee and J. K. Aggarwal, Determining vanishing points from perspective images, *Computer Vision Graphics Image Process.* 26, 256-267 (1984).
 26. B. Brillault-O'Mahony, New method for vanishing point detection, *Computer Vision Graphics Image Process. Image Understanding* 54, 289-300 (September 1991).
 27. J. B. Burns, R. S. Weiss and E. M. Riseman, View variation of point-set and line-segment features, *IEEE Trans Pattern Anal. Mach. Intell.* 15 (January 1993).

APPENDIX: PARALLEL LINES IN 3D SPACE AND THEIR PROJECTIONS

In the daily experiences, the projected lines of parallel lines in 3D space often look parallel on the image plane (within a small range of angle deviation). Of course, in some special cases, these projected image lines may look very different from each other. In this Appendix, we will explain this phenomenon quantitatively. Furthermore, this phenomenon is utilized for helping the job of identifying parallel line sets in 3D space. A similar analysis about the viewing variation of projected line segments is proposed by reference (27).

Let's consider a camera which has a small viewing field. For convenience, the image plane is assumed to be a circular area of radius a . The defined ϵ vector of any parallel lines (in 3D space, orientation vector = d) which can be observed by the camera must fall on an arc AB (on a unit sphere). It is shown in Fig. A.1. Notice the circular band region near the plane $Z=0$ is confined by radius a . The width of the band is $2 \cdot \tan^{-1} a$ rad.

The equation of a projected line l can be written as

$$\epsilon_x x + \epsilon_y y + \epsilon_z = 0. \tag{A1}$$

The projected line can also be equivalently represented by a pair (r, ϕ) shown in Fig. A.2. Therefore, a set of approximately parallel-looking on the image plane must have very similar values of angle ϕ .

Because the unit vector of $[\epsilon_x, \epsilon_y]^T$ represents the orthogonal direction ϕ of an image line, we define a new arc A'B' on the great circle lying on the $Z=0$ plane. First, the point A projects a point A'' on the $Z=0$ plane along the Z-axis.

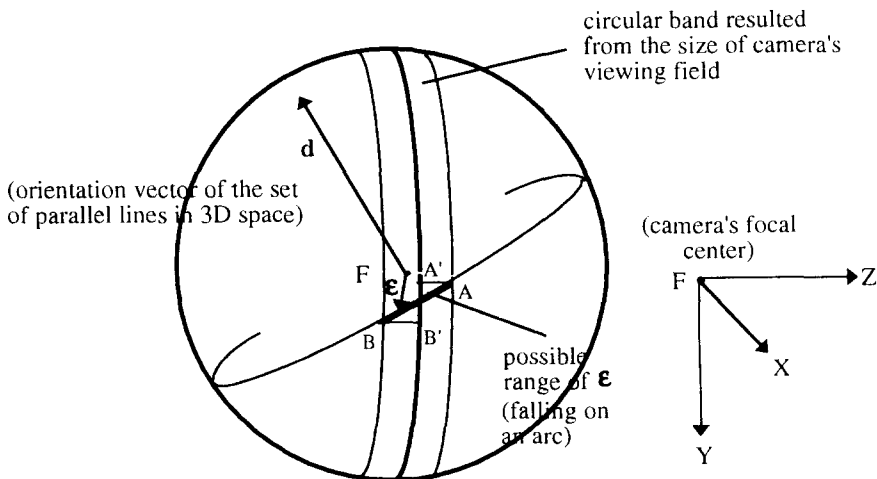


Fig. A.1. Because of the limited viewing field, the possible positions of the unit vectors ϵ must fall on an arc AB.

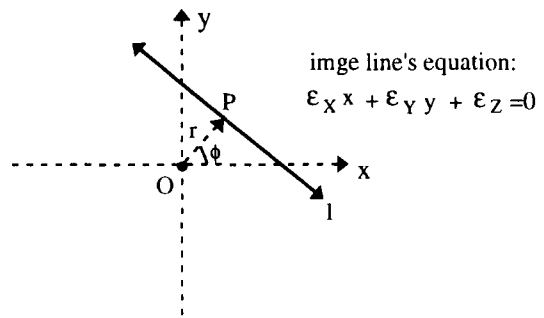


Fig. A.2. A line on the image plane can be represented by a pair (r, ϕ) . Here the vector \overline{OP} is orthogonal to this line. Obviously, the ϕ values of a set of lines which look approximately parallel on the image plane will be very close to each other.

Then the unit vector of $\overline{FA''}$ is defined as $\overline{FA'}$, similar to $\overline{FB'}$. The opening angle $\angle A'FB'$ is just the maximum possible direction deviation of the projected image lines. Without loss of generality, the orientation vector d can be defined as $[\sin \theta, 0, \cos \theta]^T$. So we have

$$\angle A'FB' = \begin{cases} 2 \sin^{-1} (|a \cdot \cot \theta|) & \text{if } \cot \theta \leq 1/a \\ \pi & \text{if } \cot \theta > 1/a, \end{cases} \tag{A2}$$

where $0 \leq \theta \leq \pi/2$ and the domain of the inverse sine function is defined in the range $(-\pi/2, \pi/2]$. Obviously, the larger the angle $\angle A'FB'$ is, more different the projected image lines look like. Two factors play important roles on it—viewing field a and line orientation d (that is θ). Figure A.3 shows the influence from these two factors. Smaller a and larger θ always induces a smaller angle deviation. Figure A.4 shows the image lines which are projected by a parallel line set with different θ values. Readers may check them with the results shown in Fig. A.3.

This concept can be utilized for helping the task of identifying the sets of parallel lines in 3D space. Because we use a camera of small viewing field, it is highly possible that a set of approximately parallel-looking lines on the image plane may correspond to the actual set of parallel lines in 3D space. So we can make an initial guess about the parallel lines. In a man-made environment, this initial guess is usually good enough. Of course, several post-tests are still needed for the final identification. They are described in the main text.

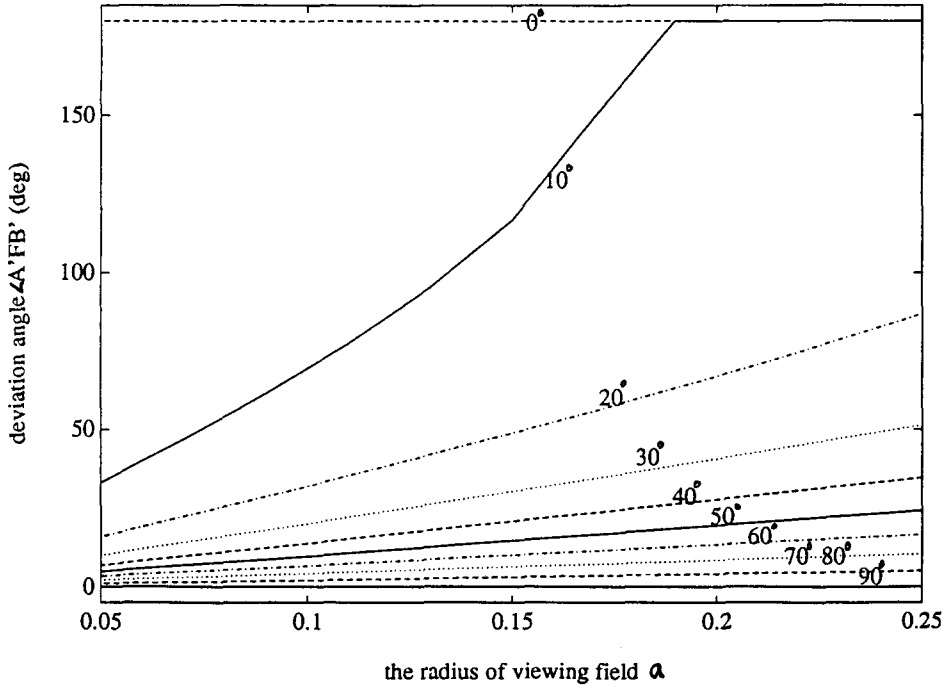


Fig. A.3. The angle deviations of the image lines which are projected by a set of parallel lines in 3D space. The angle θ varies from 0 to 90° . We know the set of parallel lines corresponding to 90° has the smallest angle deviations.

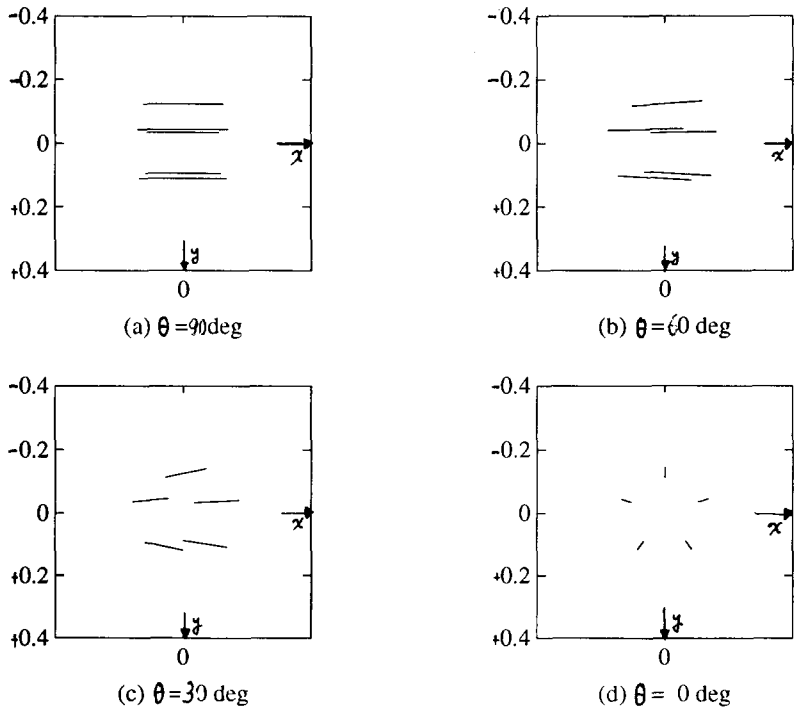


Fig. A.4. The image lines projected by the set of parallel lines corresponding to different θ ($=0, 30, 60, 90^\circ$). Readers may check it with the results shown in Fig. A.3.

About to Author—SOO-CHANG PEI was born in Soo-Auo, Taiwan, China, on 20 February 1949. He received the B.S. degree from National Taiwan University in 1970 and the M.S. and Ph.D. degrees from the University of California, Santa Barbara in 1972 and 1975, respectively, all in electrical engineering.

He was an engineering officer in the Chinese Navy Shipyard at Peng Fu Island from 1970 to 1971 and a Research Assistant at the University of California, Santa Barbara from 1971 to 1975. He was Professor and Chairman in the Department of Electrical Engineering at Tatung Institute of Technology from 1981 to 1983. He is presently the professor of the Department of Electrical Engineering at National Taiwan University. His research interests include digital signal processing, digital picture processing, optical information processing, laser and holography.

Dr Pei is a member of the IEEE, Eta Keppa Nu and the Optical Society of America.

About the Author—LIN-GWO LIU was born in Taiwan. He received the B.S. degree from National Chiao Tung University (N.C.T.U.) in Taiwan in 1989. Now he is at the National Taiwan University (N.T.U.) for his Ph.D. degree. His research interests include motion image analysis, methods for 3D object reconstruction and pattern recognition in image application.