1988

Towards Implementing Robust Visual Motion Computations: Generate and Test Approach

Chia-Hoang Lee

Report Number:
88-756

TOWARDS IMPLEMENTING ROBUST VISUAL MOTION COMPUTATIONS: GENERATE AND TEST APPROACH

Chia-Hoang Lee

CSD-TR-756
April 1988

Towards Implementing Robust Visual Motion Computations: Generate and Test Approach

Chia-Hoang Lee
Department of Computer Science
Purdue University
West Lafayette, Indiana 47907

Abstract

Despite great advances in the analysis of time-varying images, implementing or searching for correct and robust algorithms is still challenging and elusive. This paper examines some of the reasons why the motion solution to a problem is sensitive to small changes in the image data. We will discuss whether these sources of difficulty are inherent, predictable, or avoidable. We will also present a computational procedure, in contrast to previous techniques, based on a generate-and-test strategy as a first step toward implementing robust visual motion computations.

The support of the National Science Foundation under grant IRI-8702053A1 is gratefully acknowledged.

1. Introduction

Despite great advances in the analysis of time-varying images, implementing or searching for correct and robust algorithms is still challenging and elusive. This paper examines some of the reasons why motion algorithms are sensitive to small changes in the image data. We will also present a generate-and-test computational procedure as a first step toward implementing robust visual motion computations.

In measuring visual motion, it is common to distinguish two schemes. One, called the flow-based method, is based directly on local changes in light intensity. The other one, called the featured-based method, is based on identifying features. The feature-based method first segments each frame of the image sequence and marks the feature points. Next, it establishs the correspondence of these points between the two frames. Finally, it derives the motion parameters and object structure. This last step is called the structure from motion problem or simply the motion problem. Our specific interest and much of the discussion in this paper focus on the motion problem faced in the feature-based approach. For a recent survey of flow-based methods, see [5].

Two different computational procedures for motion recovery are iterative search, and linear computations followed by singular value decomposition of a matrix. In contrast to these, our proposed method adopts the generate-and-test strategy. This strategy consists of two phases: the generation phase yields plausible solutions and the testing phase selects the correct solution. In our scheme, the generation of plausible solutions is based on a pair of three non-collinear feature points. The testing checks whether a plausible solution correctly interprets the remaining features.

In the next section the problem of structure from motion is formulated and previous research is briefly reviewed: what results have been obtained, the computational issues, and a discussion of generate-and-test strategy. Section 3 briefly presents a technique employed in the generation phase. This technique solves the structure from motion for four coplanar points. A detailed discussion can be found in [13]. Section 4 discusses how a pair of sets of three points is extended to many pairs of sets of four coplanar points. Once pairs of sets o four coplanar points are formed, the generating technique described in section 3 can then be applied to derive plausible solutions. Section 5 presents a technique to test whether these plausible solutions correctly interpret the remaining points. Section 6 summarizes our scheme in a computational algorithm. Section 7 describes a series of experiments to demonstrate the potential of this technique. The last section contains discussion and conclusions.

2. The Problem of Structure from Motion

The problem of structure from motion in two views, to a certain degree, is well understood in the theoretical aspect. By the theoretical aspect we mean the number of observable features needed in the input images to recover the motion uniquely. However, the computed solution, based on known techniques, is often quite sensitive to a small amount of noise in the image data. This experience leads researchers, for instance [9][12] to employ many frames for motion recovery. This line of research represents a very promising alternative towards robust visual motion computations. Instead of increasing number of frames we investigate this problem from a so-called generate-and-test problem solving strategy. This new perspective appears to provide a robust computational scheme.

2.1 Problem Formulation

We assume that the image plane is stationary and that two perspective views at time t_1 and t_2, respectively, are taken of a N-point rigid object moving in the 3-D space. The task is to derive the motion and structure of the object in 3D space from the two views.

We shall use the following notations. The focal length f will be assumed to be 1, i.e., the image plane is at a distance one along positive z axis from the camera origin. In fact one may use f instead of 1 for the derivation of technique. Let

$z_i\, A_i$ = Object-space coordinates of a point P_i on the rigid object at t_1

$z'_i\, B_i$ = Object-space coorindates of the same point P_i at t_2

A_i = Image-space coordinates of the point P_i at t_1

B_i = Image-space coordinates of the point P_i at t_2

Notice that the third component of A_i and B_i is 1. Then

$$z'_i\, B_i = \boldsymbol{R}\, z_i\, A_i + \boldsymbol{T} \quad i = 1,..,N$$

where

$$R = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{12} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \text{ is a rotation matrix,}$$

and $T = [t_x\ t_y\ t_z]^t$ is a translation vector.

The problem we are trying to solve is: Given N image point correspondences

$$A_i <\rightarrow B_i \quad ; \quad i = 1, 2, \cdots N$$

determine R, T, and z_i, z'_i, $i = 1, \cdots N$.

If (R,T,z_i,z'_i) is a solution, then (R,cT,cz_i,cz'_i) is also a solution for any positive scalar c. Thus one could at best derive z_i, z'_i and T up to a scale factor. Figure 1 depicts the imaging geometry and the problem. If one interchanges the two frames, one obtains the following.

$$z_i A_i = R^t z'_i B_i - R^t T \quad i = 1,..,N$$

We will use S to denote R^tT. In other words, $RS = T$. The information T and S, derived in Section 3, plays an important role in the testing phase.

2.2 Previous Results: Iterative Search vs. Linear Computations

Roach and Aggarwal [2] show that five points in two views are sufficient to recover the structure and motion parameters. Their approach requires solving a system of 18 nonlinear equations with 27 variables. This method requires iterative search and a good initial guess of a solution. Nagel and Neuman [4] observe that the rotation matrix can be separated from the translation matrix. The idea stems from the observation that $Rz_iA_i \times z'_iB_i$, if not zero, defines a vector normal to a plane containing T ,where $\times$ stands for vector product. From this, a set of fourth-order polynomial equations in three unknowns (parameters of rotation) can be derived. This technique requires many fewer search dimensions than that in [2].

Tsai and Huang [14] have proposed a method to find the motion of a planar patch (containing at least four points) from 2-D perspective views. This technique consists of two steps: First, a set of eight "pure parameters" is defined. These parameters can be determined uniquely from two successive image frames by solving a set of linear equations. Then, the actual motion parameters are determined from these eight "pure parameters" by solving a sixth-order polynomial or by computing the singular value decomposition of a 3×3 matrix.

Tsai and Huang [1] investigated the problem of a curved surface patch in motion, and established two main results concerning the existence and uniqueness of solutions in the method referred to here as the 8-point algorithm. Given the image correspondences of eight object points in general position, an E matrix can be determined by solving eight linear equations. The actual 3-D motion parameters can be determined uniquely from E. If E is unique, the 3-D motion parameters are unique. A similar algorithm (not addressing the aspect of uniqueness) was also discovered independently by Longuest-Higgins [6]. In [7], he enumerates inherent configurations that lead to

singularities of E; hence 3D motion parameters are not necessarily unique, for instance, if any six of the points lie on a conic, if any four of the points are collinear, or if any seven of the points lie in a plane.

2.3 The Difficulty of Motion Computations

The computational aspect of the 8-point algorithm involves two major steps. The first step involves solving a linear system of equations. The second step involves the singular value decomposition of a matrix. It is well known that the stability of either of these two steps depends on the condition of a matrix. Therefore, the behavior of the 8-point algorithm depends on the data and can be predicted. However, according to [16], the experimental results are often sensitive to the data.

Another difficulty lies in the usage of parameters. Consider the following example: A rotation with parameters RA = (10°, 10°, 10°) , where RA denotes (rotational angle, tilt, slant), is applied to a set of eight points of which no four points are coplanar, followed by the translation (2, 2, 8). Figure 2 depicts the two input images in a 512×512 screen. In this experiment, the focal length is one, the field of view is 60°, and the spatial resolution for 1 pixel is thus about 0.00225. Since the screen coordinates are used as input, any points within 0.5 pixel of the center of the picture element (pixel) are regarded as coincident. In [15], three different solutions which match observable input images perfectly (without tolerating any pixel difference) are found and described in Table I.

As these solutions are all perfectly correct with regard to finite resolution of an image, there is no basis to favor one over the other[1] (i.e. any one of them could be used as reference parameters). The example actually raises the issue of robustness of any potential motion algorithm. If one examines the motion parameters, then the relative error in tilt might be as large as 400% and the relative error in slant might be 70%. However, the use of relative error is quite mesleading because small angles will inevitably cause large relative errors. If absolute error is used, then the error in tilt might be 40° where the range of tilt is 360°, thus a 10% error. If one examines other parameters, the example shows errors up to 10%. This points out the difficulties encountered in [1] and any existing algorithm. However, we have also observed [15] that there is a closeness between the derived matrix and the actual matrix. Therefore we suggest that the rotational matrix be used for comparison.

2.4 A Generate-and-Test Computational Strategy

[1] While the actual reference data satisfies the assumption of the uniqueness theorem in [1], there is no contradiction in having many solutions. In the case of finite resolution, one could create many input images (in terms of infinite resolution) having the same two finite resolution input images. Thereafter, one could recover motion parameters for each set of input images; hence one presumably may anticipate quite a few solutions.

Consider an N-point motion problem. The mechanism of most algorithms is to take all N points into account (i.e., none of them will be left out) and generate a unique solution in "one" step. Such an idea has yet to support a robust algorithm that would respond to noise gracefully for N less than 20 points. It is, however, evident that the more feature points a technique requires, the fewer applications it has.

In our two-step approach, the first objective is to generate plausible solutions to account for a pair of sets of three image points irrespective of other features. The second objective is to test these generated solutions to see if any of them could account for points not considered in the first place. See Figure 3 for a sketch of these two paradigms. For a problem of N points, we have N(N-1)(N-2)/6 possible groupings of the points in threes. While we could generate plausible solutions based on one grouping, we generate them from all possible groupings instead. It is likely that the position of some of the points would be very accurate. Hence the plausible solutions generated by the group consisting of these accurate points would contain the correct one. When one tests this correct solution against other points, it would presumably be within the acceptable range. Note that when a plausible solution is tested against the other points, one should not anticipate a perfect match because we are dealing with noisy images. Thus one has to design some criterion of tolerance. Since many of them could qualify as solutions, a question arises: Do they form a cluster? If the problem is of size greater than eight feature points, then there must exist a cluster because the motion is continuous. From the cluster, one has to choose a representive. The correctness of the algorithm is ensured by the theoretical basis established in [1] for more than eight feature points. The difference of this strategy from iterative search is that all the candidates are generated and examined one by one. One advantage of our approach is that the theoretical error analysis is easier to establish. Another advantage is its redundancy which usually provides the hope of achieving robustness in computations.

3. Basic Problem

In this section, we will introduce a basic problem which plays a central role in the generation phase. This basic problem can be defined as a motion problem for four coplanar points $z_1A_1, z_2A_2, z_3A_3, z_4A_4$ subject to the following constraints: A_1, A_2, A_3 is not collinear in the image; and B_1, B_2, B_3 (correspondence of A_i in the second image) is not collinear in the image; and A_4 is the midpoint of the triangle defined by A_1, A_2, A_3.

A general problem of a four-point coplanar patch was solved in [14] and reexamined in [13]. For the basic problem, either technique could be used. The former method requires first solving linear equations and then finding the singular value of a matrix. These two steps are not necessarily well-conditioned. The latter method also involves solving linear equations and then finding the eigenvalues of a positive definite matrix. While the first step of solving linear equations is not necessarily well-conditioned, the

second step is always well conditioned. For this reason, we have chosen the latter technique and describe it briefly below in Section 3.1 and 3.2. For details of this approach, see [13].

3.1 DERIVING THE RATIO OF DEPTHS

As we shall see presently, by using planarity, we can determine all z_i/z'_i up to a unknown constant k. This observation is based on the following. Recall that

$$\boldsymbol{R}\, z_1\, A_1 = z'_1\, B_1 - T \tag{1}$$

$$\boldsymbol{R}\, z_2\, A_2 = z'_2\, B_2 - T \tag{2}$$

$$\boldsymbol{R}\, z_3\, A_3 = z'_3\, B_3 - T \tag{3}$$

$$\boldsymbol{R}\, z_4\, A_4 = z'_4\, B_4 - T \tag{4}$$

Since $z_4\, A_4$ is coplanar with $z_1\, A_1$, $z_2\, A_2$, $z_3\, A_3$, we know that there exists a_1, a_2, a_3 such that

$$a_1 + a_2 + a_3 = 1$$

and

$$a_1\, z_1\, A_1 + a_2\, z_2\, A_2 + a_3\, z_3\, A_3 = z_4\, A_4. \tag{5}$$

Applying $\boldsymbol{R}$ to both sides of (5) and using (1)-(4), one could rewrite it as

$$a_1\, z'_1\, B_1 + a_2\, z'_2\, B_2 + a_3\, z'_3\, B_3 = z'_4\, B_4. \tag{6}$$

Dividing both sides of (5) and (6) by z_4 and z'_4 respectively, one gets

$$a_1 \frac{z_1}{z_4} A_1 + a_2 \frac{z_2}{z_4} A_2 + a_3 \frac{z_3}{z_4} A_3 = A_4 \tag{7}$$

and

$$a_1 \frac{z'_1}{z'_4} B_1 + a_2 \frac{z'_2}{z'_4} B_2 + a_3 \frac{z'_3}{z'_4} B_3 = B_4. \tag{8}$$

Since A_4 is the midpoint among A_1, A_2, A_3, it is obvious that

$$a_1 \frac{z_1}{z_4} = \frac{1}{3};\; a_2 \frac{z_2}{z_4} = \frac{1}{3};\; a_3 \frac{z_3}{z_4} = \frac{1}{3}. \tag{9}$$

Rewriting (8) in matrix form, we have

$$[B_1\ B_2\ B_3]\,[q_1\ q_2\ q_3]^t = B_4 \tag{10}$$

where

$$a_1\ \frac{z'_1}{z'_4} = q_1\ ;\ a_2\ \frac{z'_2}{z'_4} = q_2\ ;\ a_3\ \frac{z'_3}{z'_4} = q_3.$$

By simple calculations, one gets

$$\frac{z'_2}{z_2} = \frac{q_2}{q_1}\ \frac{z'_1}{z_1} = \delta_2\ \frac{z'_1}{z_1}\ \text{ and }\ \frac{z'_3}{z_3} = \frac{q_3}{q_1}\ \frac{z'_1}{z_1} = \delta_3\ \frac{z'_1}{z_1}.$$

Now we denote $\dfrac{z'_1}{z_1}$ by k. Then $z'_2/z_2 = \delta_2\ k$ and $z'_3/z_3 = \delta_3\ k$ where δ_2 and δ_3 are defined as above.

3.2 DERIVING MOTION PARAMETERS

Once δ_2,δ_3 are obtained from the previous section, the following three theorems recover the translational vector T in different cases. [13] contains the details of the theorems and also includes a simple formula for the rotational matrix. In our implementation, the rotational matrix is derived in section 4.2 based on the other method instead. Let

$$A^t = [B_1\ \ \delta_2 B_2\ \ \delta_3 B_3]\,[A_1\ A_2\ A_3]^{-1}$$

Theorem 1:

If the eigenvalues of $A^t\ A$ are all equal, then $T = 0$.

Theorem 2:

If the three eigenvalues, λ_1, λ_2, λ_3, of $A^t\ A$ are such that $\lambda_1 = \lambda_2 \neq \lambda_3$ and v_1, v_2, v_3 are associated eigenvectors, then $T = v_3$.

Theorem 3:

If the three eigenvalues λ_1, λ_2, λ_3 of $A^t\ A$ are distincit and v_1, v_2, v_3 are associated eignvectors and $\lambda_1 > \lambda_2 > \lambda_3$, then $T = v_1 \pm \varepsilon\ v_3$ where $\varepsilon = \sqrt{(\lambda_2-\lambda_3)/(\lambda_1-\lambda_2)}$.

Recall that if we reverse the order of the frames, then S, which describes the translation of the motion from the second frame to the first frame, can also be recovered in the same way T was derived.

4. Deriving Plausible Solutions: Generation Phase

Consider two perspective images of N points. It is apparent that any three points in the 3D-space are coplanar. Our strategy now is to seek a way to apply the technique for the basic problem to any three noncollinear image points.

To illustrate this, consider a pair of sets of three points, A_1,A_2,A_3 and B_1,B_2,B_3. First we will introduce the midpoint of triangle $\Delta A_1A_2A_3$ as the fourth point A_4 and demand z_4A_4 to be coplanar with z_1A_1,z_2A_2,z_3A_3. In a word, we regard A_4 as the projection of a point lying on the plane defined by $z_1A_1,z_2A_2,\ z_3A_3$. Although z_4A_4 is purely an artifact, it facilitates the expansion of a set of three points to a set of four points. One immediate question is where the corresponding point of A_4, denoted by B_4, is in the second image. Because of nonlinearity, the location of B_4 is at best known to lie inside the triangle $\Delta B_1B_2B_3$. Thus the location of B_4 depends on the number of pixels within the triangle B_1,B_2,B_3. By using pixels (finite resolution) and allowing a 0.5 pixel error, we reduce the number of cases from infinitely many to finitely many. For each pixel within $\Delta B_1B_2B_3$, we formulate an instance of a basic problem corresponding to the pair of A_1,A_2,A_3,A_4 and B_1,B_2,B_3,B_4. Once a pair of four points is created, the technique for the basic problem may be applied to derive T and S. Notice that there may be up to four pairs of (T,S) by Theorem 3 of Section 3.2.

5 Testing Plausible Solution

As pointed out above, each possible pair of sets of three points will generate up to four times as many $(T,S)'s$ as there are pixels in one of the triangle. The testing phase uses the remaining image points (i.e., A_i,B_i) to prune these (T,S)[2]
To do this, we first show that: (i) $\{A_i \times S : i = 1, N\}$ and $\{B_i \times T : i = 1, N\}$ are coplanar, respectively and (ii) the angle between $A_i \times S$ and $A_j \times S$ equals the angle between $B_i \times T$ and $B_j \times T$

Clearly, $A_i \times S$ is a vector lying on a plane perpendicular to vector S, and $B_i \times T$ is a vector lying on a plane perpendicular to T. Thus $\{A_i \times S : i = 1, N\}$ and $\{B_i \times T : i = 1, N\}$ can be regarded as two patterns of N coplanar rays emanating from the origin. To show that the angle between $A_i \times S$ and $A_j \times S$ equals the angle between $B_i \times T$ and $B_j \times T$, recall that $Rz_i\ A_i = z'_i\ B_i - T$ and $RS = T$. Thus

$$z_i\ R(A_i \times S) = z'_i(B_i \times T)$$

From this, one can conclude that the angles must be the same.

---

[2] One could also derive R based on eigenvectors by theorems of section 3.2. Once both R and T are available, one can substitute them into motion equations. This will lead to three linear equations in terms of z_i and z'_i. Because of noise, these three lines do not intersect, in general, at a comon point. Thus the next step is to design some criterion of acceptance. A possible criterion could use the area bounded by the line segments connecting three intersections.

To obtain the depth z_i, observe that

$$\frac{z_i}{z'_i} = \frac{||B_i \times T||}{||A_i \times S||}$$

Furthermore,

$$z_i A_i = -T \cdot T + z'_i B_i \cdot T$$

Using the above, simple calculations give us

$$z_i = \frac{||B_i \times T|| \, ||T||}{||A_i \times S|| \, (B_i \cdot T) - ||B_i \times T|| \, (A_i \cdot S)}$$

Now our testing strategy is (1) Compute the angle, denoted by α_i, between $A_i \times S$ and $A_{i+1} \times S$ and the angle, denoted by β_i, between $B_i \times T$ and $B_{i+1} \times T$. (2) Require the difference $|\alpha_i - \beta_i|$ to be a small angle. In our experiments, we considered two degrees to be small. (3) Accept a solution if it passes through (1) and (2) and compute R by the following:

$$R : S \longrightarrow T$$

$$R : A_1 \times S \longrightarrow B_1 \times T$$

$$R : S \times (A_1 \times S) \longrightarrow T \times (B_1 \times T)$$

(4) Compute z_i and require z_i be positive. (5) Compute $Rz_i A_i + T$. (6) Compute the screen coordinates $\hat{B}_i$ based on (5). (7) Require the difference between $\hat{B}_i$ and B_i to be as small as possible. In general, we require that all the points should be within a reasonable pixel-tolerance, e.g. 4 pixels, and a majority of the points should be within a very strict pixel-tolerance, e.g. one pixel. If a solution passes these tests, then we accept it.

6. Algorithm

The following algorithm summarizes our method.

Algorithm:

(*Input Images*):
Two Frames: A_i i=1..N and B_i i=1..N /* screen images */
For each i<j<k such that $\{A_i, A_j, A_k\}$ are noncollinear and $\{B_i, B_j, B_k\}$ are nocollinear

cobegin:
for each pixel B inside $\Delta B_i B_j B_k$
cobegin:
(1) solve $[B_i\ B_j\ B_k]\ Q = B$ where $Q = [q_1\ q_2\ q_3]$ and let $\delta_2 = q_2/q_1$; $\delta_3 = q_3/q_1$
(2) compute the eigenvalues and eigenvectors of
$[B_i\ \delta_2 B_j\ \delta_3 B_k][A_i\ A_j\ A_k]^t [A_i\ A_j\ A_k]^{-t}\ [B_i\ \delta_2 B_j\ \delta_3 B_k]^t$
compute $\boldsymbol{T}$
(3) compute the eigenvalues and eigenvectors of
$[A_i\ A_j/\delta_2\ A_k/\delta_3][B_i\ B_j\ B_k]^t [B_i\ B_j\ B_k]^{-t}\ [A_i\ A_j/\delta_2\ A_k/\delta_3]^t$
compute $\boldsymbol{S}$
(3) compute
α_i: angle between $A_1 \times \boldsymbol{S}$ and $A_i \times \boldsymbol{S}$
β_i: angle between $B_1 \times \boldsymbol{T}$ and $B_i \times \boldsymbol{T}$
(4) if $|\alpha_i - \beta_i| <$ *degree_tolerance* for every i, then compute $\hat{R}$ based on
$\hat{R}: \boldsymbol{S} \longrightarrow \boldsymbol{T}$
$: A_i \times \boldsymbol{S} \longrightarrow B_i \times \boldsymbol{T}$
$: A_i \times (A_i \times \boldsymbol{S}) \longrightarrow \boldsymbol{T} \times (B_i \times \boldsymbol{T})$

(5) compute $z_i = \dfrac{\|B_i \times \boldsymbol{T}\| \, \|\boldsymbol{T}\|}{\|A_i \times \boldsymbol{S}\| \, (B_i \cdot \boldsymbol{T}) - \|B_i \times \boldsymbol{T}\| \, (A_i \cdot \boldsymbol{S})}$
(6) use $\hat{R}$, $\hat{z}_i$, A_i to compute $\hat{B}_i$.
(7) If $|\hat{B}_i - B_i| <$ pixel_tolerance for every i, then accept the solution.
coend
coend

7. Experiments

The observable input images are all in screen coordinates. In other words, simulations are based on finite resolution images. By using finite resolution, a 0.5 pixel error is already inherent in the position of each feature. As to the parameters of the imaging geometry, the field of view of the camera is 60°, the focal length is one, and the image is 512×512 pixels. Three experimental conditions are considered: (I) a pair of finite resolution images; (II) randomly add zero to two pixels noise in the x and y directions to the images generated in (I) and average over 20 samples; (III) randomly add zero to four pixels noise in the x and y directions to the images generated in (I) and average over 20 samples. The results are presented in terms of the relative error in T, the motion parameters, and a measure described below.

Note that the following holds for any x:

$$\frac{\| \hat{R}\, x - R\, x \|}{\| x \|} \leq \| \hat{R} - R \|$$

The geometric meaning is that the angle between $R\,x$ and $\hat{R}\,x$ (for every x) cannot exceed $2 \sin^{-1} \left(\frac{\| R - \hat{R} \|}{2} \right)$. This fact can be seen in Figure 4 where $\| R - \hat{R} \|$ is the length defined by $R\,x$ and $\hat{R}\,x$. We use this as a measure[3].

(I) Finite Resolution:

A set of eight points, of which none of the four points are coplanar, were chosen arbitrarily. We then applied a motion with 10° of tilt, 10° of slant, and 10° of rotational angle to these points followed by a translation given by the vector (2,2,8). We denote the rotation parameters as (tilt, slant, rotational angle). In other words, the rotation parameters were (10°,10°,10°). We used 2° as the tolerance in step 4 of the above algorithm and 2 pixels as the tolerance of mismatch for each point on the second image. Seventy solutions passed the testing phase. Of all these 70 solutions, the error incurred in the rotational matrx never exceeded 1°. In other words, a cluster is formed. The result is described in the first row of the Table A.

(II) Noise in both x and y components: Zero to Two pixels.

The x and y components of each feature point in the above input images was corrupted by zero, one, or two pixels randomly. The second row of Table A gives the averages of the derived parameters in 20 experiments.

(III) Noise in both x and y components: Zero to Four pixels

[3] The technique used to estimate the norm $\|R-\hat{R}\|$ is based on Gerschgorin's circle theorem ([8]) for eigenvalue and the theorem the norm of A is the square root of the largest eigenvalue of A^TA ([8]).

This case repeated (II) except that the noise ranged from zero to four pixels. The third row in Table A gives the results over 20 experiments.

We also conducted all the above three experiments using two other sets of motion parameters (10°,30°,20°) and (20°,30°,40°). The results are tabulated in the Table B and the Table C respectively. Notice that the rotational angle around the rotational axis in our simulations ranged from 10° to 40° and the slant, the angle between the rotational axis and the optical axis, ranged from 10° to 30°. Those represent large rotation angles relative to those allowed in [16], in which the success of the experiments relies on a small rotation angle, typically below 5°.

8. Conclusions

The experiments described above confirm that a generate-and-test strategy can be used successfully for implementing robust visual motion computations. The rotational matrix can be determined very accurately even though the noise level is increased up to four pixels. The uniqueness of the motion parameters will require at least eight features in each of the two images. The mean of the cluster of acceptable motions is chosen as the final solution. In our current implementation, it takes a substantial amount of cpu time to compute the solution. However, the algorithm could be implemented much more efficiently on a parallel machine. Some future questions: (i) Could one have a basic problem that would reduce the amount of computation? (ii) Could a least-squares method be incorporated into the computation process?

References

1. R.Y. Tsai and T.S. Huang, "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces", *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol PAMI-6, No. 1, Jan. 1984.
2. J.W. Roach and J.K. Aggarwal, "Determining the Movement of Objects from a Sequence of Images", *IEEE Trans. Pattern Anal. Machine Intelligence*, Vol. PAMI-2, Nov. 1979.
3. S. Ullman, *The Interpretation of Visual Motion*, Cambridge, MA, MIT Press, 1979.
4. H.H. Nagel and B. Neumann, "On 3-D Reconstruction from Two Perspective Views", in *Proc. IJCAI* 81, Vol. II, Aug. 1981.
5. H.H. Nagel, "Image Sequences - Ten (octal) years - from phenomenology towards a theoretical foundation Pattern Recognition, Paris, France (1986) 1174-1185
6. H.C. Longuet-Higgens, "A Computer Algorithm for Reconstructing a Scene from Two Projections", *Nature*, 293, 1981.
7. H.C. Longuet-Higgins, "The Reconstruction of a Scene from Two Projections – Configurations that Defeat the 8-Point Algorithm", *Proc. of the First Conf. on Artificial Intelligence Applications*, Dec. 1984.
8. Strang, Gilbert, *Linear Algebra and Its Applications 1980*, Academic Press, p. 288-304 New York.
9. G.S. Young and Rama Chellappa, *3-D Motion Estimation Using a Sequence of Noisy Stereo Images*, CVPR, Michigan Ann Arbor, 1988.
10. J.K. Aggarwal, "Motion and time-varying imagery-An overview. Proc. of IEEE Workshop on Motion: Representation and Analysis, South Carolina, 1986.
11. J.K. Aggarwal and A. Mitiche "Structure and Motion from Images: facts and fiction." Proc. of IEEE Workshop on Motion: Representation and Analysis, South Carolina, 1986.
12. J. Weng et.al "3D motion estimation, understanding, and prediction from noisy image sequence. IEEE PAMI Vol-PAMI-9, pp. 370-389, May 1987.
13. C.H. Lee, "On Motion of a Rigid Planar Patch from Perspective projections" Technical Report, Department of Computer Sciences, Purdue University (1988).
14. R.Y. Tsai, T.S. Huang and W. Zhu, "Estimating Three-Dimensional Motion Parameters of a Rigid Planar Patch, II: Singular Value Decomposition", *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. ASSP-30, No. 4, 1982.
15. C.H. Lee, "Time-Varying Images: The Effect of Finite Resolution on Uniqueness" Technical Report, Department of Computer Science, Purdue University (1988).

16. J. Fang and T.S. Huang, "Some Experiments on Estimating the 3D Motion Parameters of a Rigid Body from Two Consecutive Image Frame" *IEEE Trans. Pattern Recognition and machine Intelligence*, Vol. PAMI-6, No. 5, 1984.

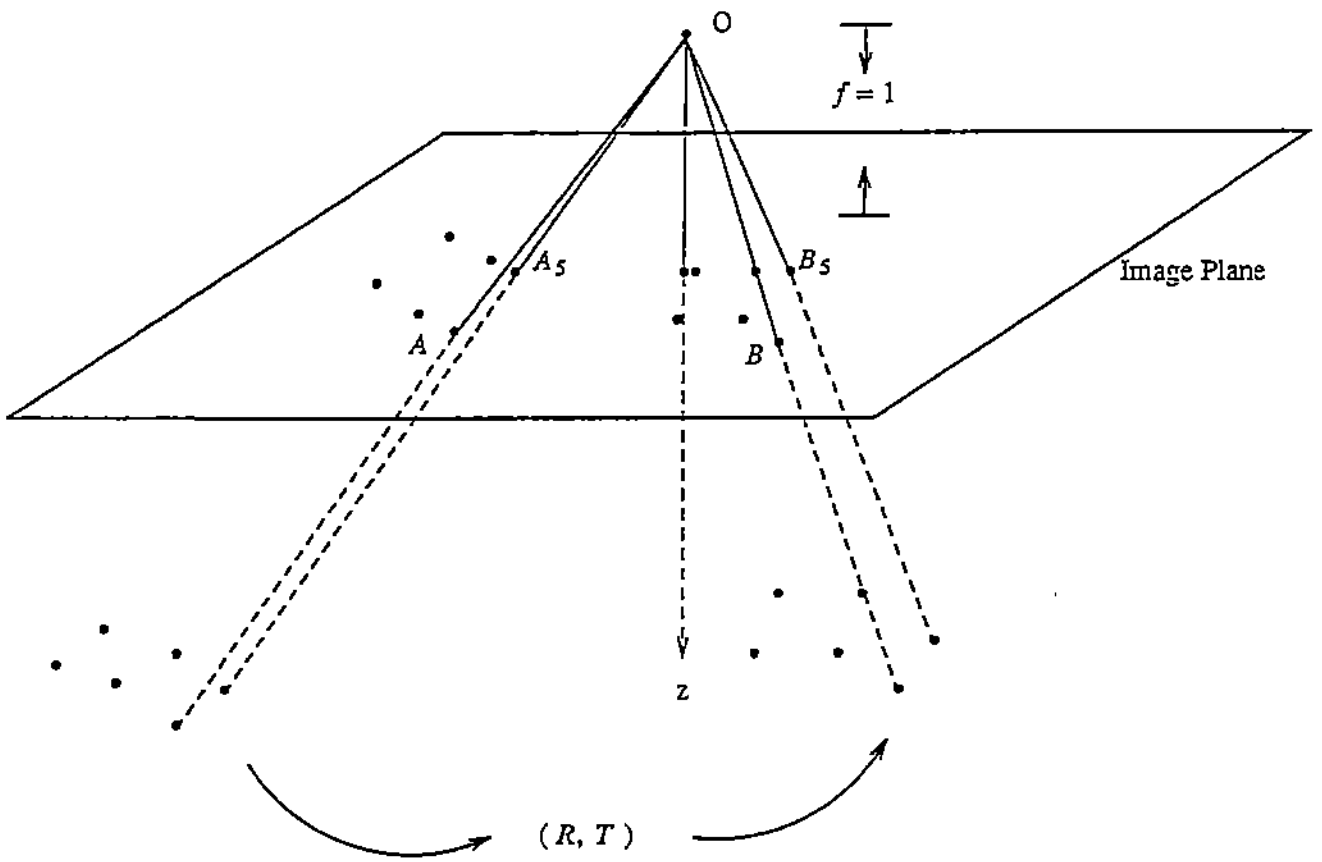

Figure 1. Imaging geometry and the problem.

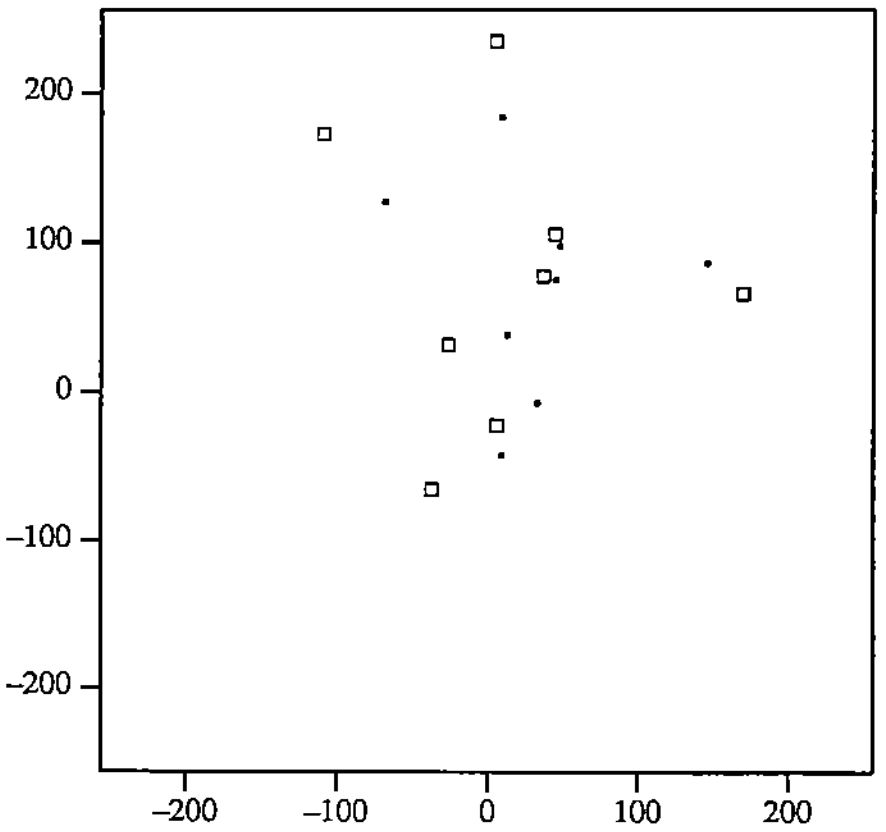

Figure 2. Squares represent the first image while dots represent the second image

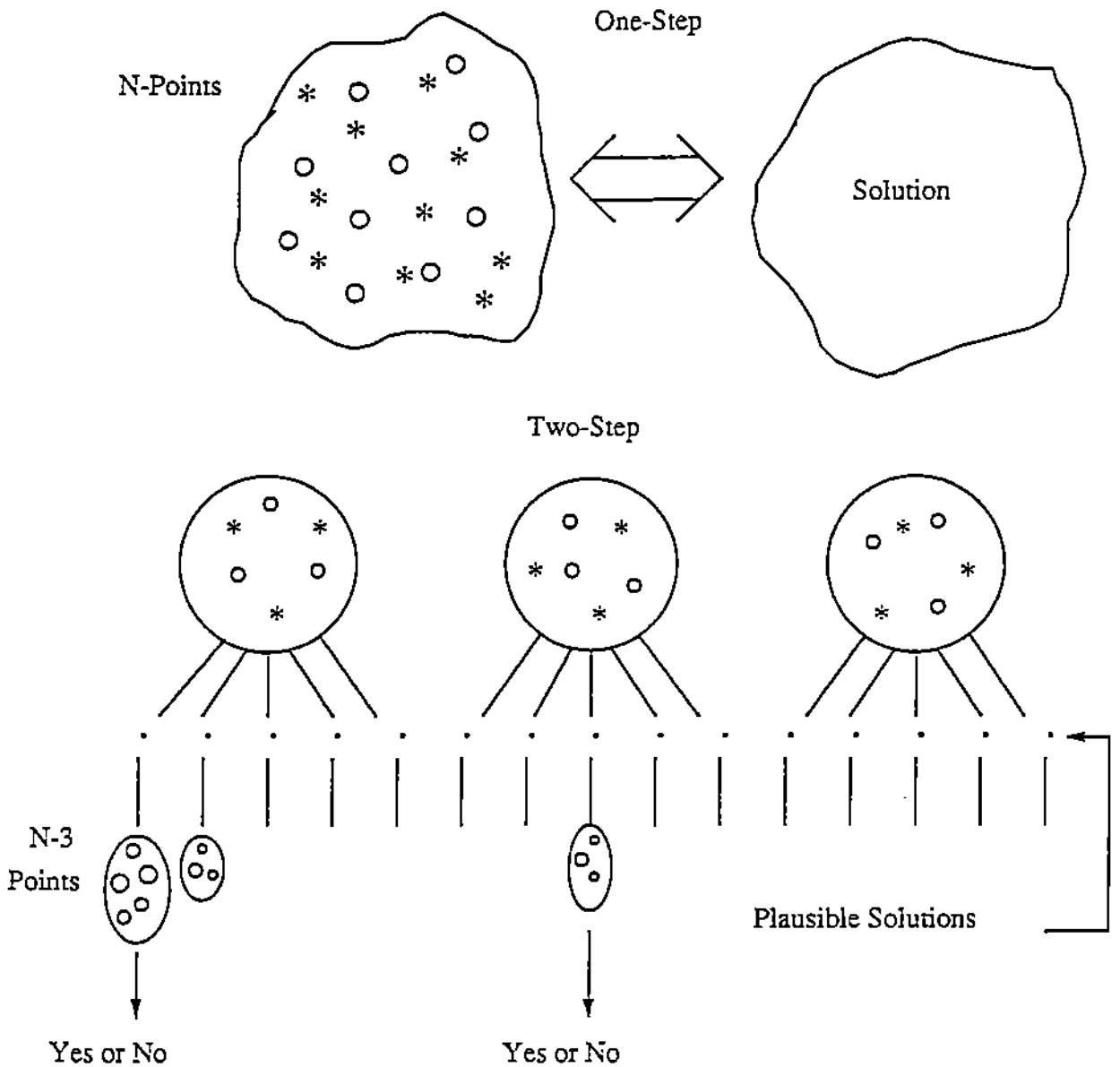

Figure 3: O's represents the first image, *'s represents the second image

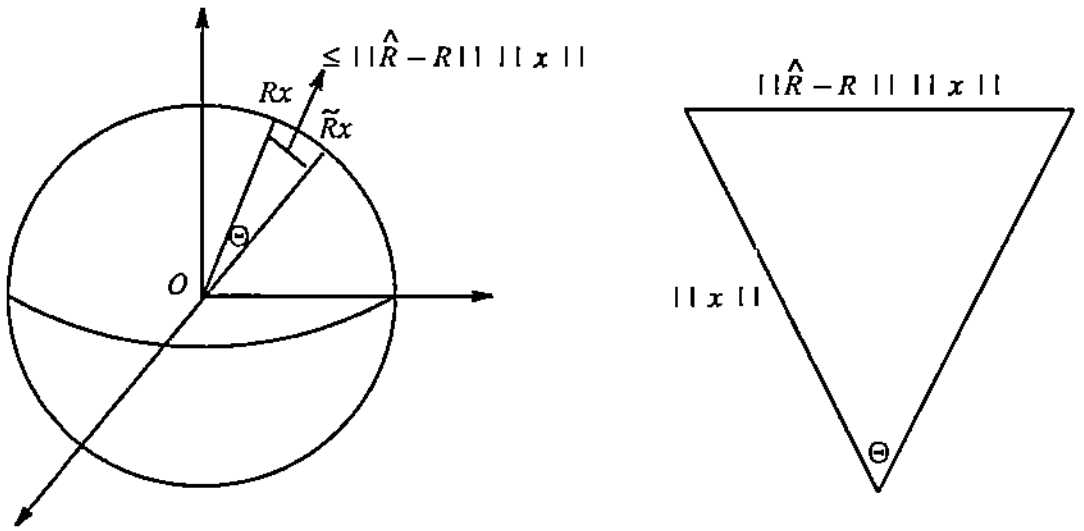

Figure 4

Table I:

tx	ty	tz	Tilt	Slant	Angle
0.281830	0.246854	1.0	-10.877623	9.363212	10.024160
0.362555	0.311039	1.0	-35.339671	17.536793	10.870609
0.25	0.25	1.0	10	10	10

Table A: RA = (10°,10°,10°), T=(2,2,8)

Noise	ER	ET	Tilt	Slant	Angle
0.5	1.02	7%	-17	14	10
0-2	1.52	10%	-9	19	10
0-4	1.63	13%	-4	16	10

Table B: RA = (10°,30°,20°), T=(2,2,8)

Noise	ER	ET	Tilt	Slant	Angle
0.5	1.13	1%	13	30	20
0-2	1.90	11%	8	30	20
0-4	3.07	21%	8	31	21

Table C: RA = (20°,30°,40°), T=(2,2,8)

Noise	ER	ET	Tilt	Slant	Angle
0.5	0.90	1%	22	29	40
0-2	3.33	17%	23	29	39
0-4	3.96	22%	26	29	40