1988

Motion Estimation of a Rigid Planar Patch: A Robust Computational Technique

Chia-Hoang Lee

Report Number:
88-746

MOTION ESTIMATION OF A RIGID PLANAR PATCH: A ROBUST COMPUTATIONAL TECHNIQUE

Chia-Hoang Lee

CSD-TR-746
March 1988
Revised March 1989

MOTION ESTIMATION OF A RIGID PLANAR PATCH: A ROBUST COMPUTATIONAL TECHNIQUE

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

ABSTRACT

Although theories for recovering motion/structure of a planar patch are known, existing algorithms are quite sensitive to the noise in the image data. This paper revisits the problem and presents robust computing procedures for recovering motion parameters. Specifically, two new concepts are introduced: (1) A virtual plane that is normal to the translational vector, and (2) the ratio of depth of a featured point over time. These two concepts drastically simplify the analysis and lead to a simple and robust procedures for recovering motion parameters. The robustness of the technique is discussed for each step involved. Extensive simulations consisting of twelve sets of different motion parameters are used to test the technique and the results using real images are also provided.

1. INTRODUCTION

One of the fundamental task in computer vision is the analysis of a sequence of imaged scene or objects. Figures 5a-5b show snapshots of a stationary scene imaged by a moving camera and the sequence in figure 6a-6b show two snapshots of a moving truck imaged by a stationary camera. A general aim of analyzing these scenes is to recover the structure of the scene/object and the underlying motion it exercises with respect to the visual sensor.

Two schemes are often used for measuring visual motion. One, called flow-based method, is based directly on the local changes in light intensity values. The other one, called feature-based method, is based on identifying features. Both schemes receive extensive attention from researchers. In their pioneering work on motion estimation of a rigid planar patch, Tsai and Huang [1] (referred to T-H method) showed that two views of a patch are sufficient to determine the motion and its structure provided that four features could be identified. However, the computational aspect of T-H method is quite sensitive to the noise in the image data and only proves to be satisfactory if many points are available. Such a least square approach requires many points and therfore restricts potential application domains of this technique. In fact, none of computing procedures based on minimum information proves to be robust.

In this paper, we show that a robust computing procedure based on minimum information can be developed. The characteristic of this procedure consists of two new concepts: (i) The motion of a virtual planar patch that is normal to the translational vector, and (ii) The concept of the ratio of the depth over time for every featured point. It turns out that these two concepts drastically simplify the analysis and provides a

robust computation for motion recovery.

The paper is organized as follows. Section 2 formulates the motion problem of a planar patch and section 3 discusses the previous work. In section 4, we introdue the concept of depth-ratio of a feature point over time and give a formula to determine it. In section 5, we derive the motion parameters based on the use of a conceptual plane that is normal to the translation vector. Section 6 summarizes each computational step in our algorithm and examines its sensitivity to the changes in the data. Section 7 discusses how the technique can be used for more than four points. Section 8 gives extensive simulation results under a variety of different motion parameters and real image demonstration. The last section contains conclusions and discussion.

2. MOTION PROBLEM FORMULATION: PLANAR PATCH

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 rigid planar patch moving in the 3-D object space. The task is to recover the motion and structure of the planar patch in 3D space from the two views. This formulation is also applicable for the case when a static scene is imaged by a moving camera.

We shall use the following notations. The focal length f will be assumed to be 1 which implies that 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

$$z'_i\, B_i = \boldsymbol{R}\, z_i\, A_i + \boldsymbol{T} \quad i = 1,..,4$$

where

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

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

The problem we are trying to solve is: Given 4 image point correspondences

$$A_i <\!\!\rightarrow B_i\ ; \quad i = 1, 2, \cdots 4$$

where $z_i\ A_i$ are coplanar, determine $\boldsymbol{R}$, $\boldsymbol{T}$, and z_i, z'_i.

It is noted that if $(\boldsymbol{R},\boldsymbol{T},z_i,z'_i)$ is a solution then $(\boldsymbol{R},c\boldsymbol{T},cz_i,cz'_i)$ is also a solution where c is a scalar. Thus one could at best derive z_i, z'_i and $\boldsymbol{T}$ up to a scale. Figure 1 depicts the imaging geometry and the problem. We also assume that no three of $A_i's$ or $B_i's$ are collinear for $i = 1,2,3,4$. In other words, the camera origin is not on the plane defined by the planar patch. Thus the planar patch should form a quadrilateral in the

image plane.

3. Previous Work

In view of the large volume of literature dealing with motion problems, we will not attempt any comprehensive review here. Interested reader may see [1][2][3] for further references.

T-H method provides an important theoretical and computational framework for the motion recovery of a planar patch in the two frames. Since our interest here is a revisit of this problem based on minimum information, we will summarize T-H technique to the extent so that the two techniques can be compared at both the conceptual and the computational level.

T-H method first describes the space-coordinates of the features in terms of the orientation of the planar patch. Next, it shows that the image-space coordinates (X_i,Y_i) before and (X_i',Y_i') after motion are related by eight linear equations of eight "pure parameters" as listed below.

$$\begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -X_1X'_1 & -Y_1X'_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -X_1Y'_1 & -Y_1Y'_1 \\ & & & & \cdot & & & \\ & & & & \cdot & & & \\ X_4 & Y_4 & 1 & 0 & 0 & 0 & -X_4X'_4 & -Y_4X'_4 \\ 0 & 0 & 0 & X_4 & Y_4 & 1 & -X_4Y'_4 & -Y_4Y'_4 \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_7 \\ a_8 \end{bmatrix} = \begin{bmatrix} X'_1 \\ Y'_1 \\ \cdot \\ \cdot \\ X'_4 \\ Y'_4 \end{bmatrix}$$

The formula for these eight pure parameters a_i are rational functions of motion parameters (there are six of them) and the orientation of the planar patch. It first concludes that there is a unique solution to this system of equations by using Lie group theory. Then it shows the singular values of the following matrix is related to the motion parameters.

$$\begin{bmatrix} a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 \\ a_7 & a_8 & 1 \end{bmatrix}$$

As for the computational aspect of T-H method, it was well known this method is extremely sensitive to the noise in the image data. Further discussion of computing issue is left in the section 6 where the robustness of our technique will be investigated and compared.

4. DERIVING THE RATIO OF DEPTHS

A new concept introduced in this section is the depth-ratio of a feature point over time. Based on our notation, this entity is denoted by z_i/z'_i for the ith feature where z_i and z'_i are, respectively, depths of the same feature at two consecutive time instances. We shall see in the next section that the depth-ratios simplify the analysis drastically. Our aim at present is to give a computing procedure for z_i/z'_i. As we shall see, by using planarity, all z_i/z'_i can be determined up to a unknown constant k. Recall that

$$\boldsymbol{R}\, z_1\, A_1 = z'_1\, B_1 - \boldsymbol{T} \tag{4.1}$$
$$\boldsymbol{R}\, z_2\, A_2 = z'_2\, B_2 - \boldsymbol{T} \tag{4.2}$$
$$\boldsymbol{R}\, z_3\, A_3 = z'_3\, B_3 - \boldsymbol{T} \tag{4.3}$$
$$\boldsymbol{R}\, z_4\, A_4 = z'_4\, B_4 - \boldsymbol{T} \tag{4.4}$$

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

$$a_1 + a_2 + a_3 = 1$$

and

$$a_1 z_1 A_1 + a_2 z_2 A_2 + a_3 z_3 A_3 = z_4 A_4 \quad . \tag{4.5}$$

Applying $\boldsymbol{R}$ to both sides of the above equation and using (4.1)-(4.4), one could rewrite (4.5) as

$$a_1 z'_1 B_1 + a_2 z'_2 B_2 + a_3 z'_3 B_3 = z'_4 B_4 \tag{4.6}$$

Dividing both sides of (4.5) and (4.6) by z_4 and z'_4 respectively, one gets

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

and

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

Let $A_{ij} = A_i \times A_j$; $B_{ij} = B_i \times B_j$ where $\times$ denotes vector product, $i, j = 1, \cdots 3$ and $i < j$.

It is easy to see that

$$a_k \frac{z_k}{z_4} A_k \cdot A_{ij} = A_4 \cdot A_{ij} \tag{4.7}$$

$$a_k \frac{z'_k}{z'_4} B_k \cdot B_{ij} = B_4 \cdot B_{ij} \tag{4.8}$$

where $k \neq i$, $k \neq j$, $k = 1, 2, 3$, and $i < j$.

Since no three of A_1, A_2, A_3 and A_4 are collinear, one could divide (4.8) by (4.7) and obtains

$$\frac{z'_k}{z_k} = \frac{A_k \cdot A_{ij}}{B_k \cdot B_{ij}} \frac{B_4 \cdot B_{ij}}{A_4 \cdot A_{ij}} \frac{z'_4}{z_4}$$

for $k = 1, 2, 3$, and $k \neq i, j$.

Following simple calculations, one gets

$$\frac{z'_2}{z_2} = \frac{A_2 \cdot A_{13}}{B_2 \cdot B_{13}} \cdot \frac{B_4 \cdot B_{13}}{A_4 \cdot A_{13}} \cdot \frac{B_1 \cdot B_{23}}{A_1 \cdot A_{23}} \cdot \frac{A_4 \cdot A_{23}}{B_4 \cdot B_{23}} \frac{z'_1}{z_1} \stackrel{def}{\equiv} \delta_2 \frac{z'_1}{z_1}$$

$$\frac{z'_3}{z_3} = \frac{A_3 \cdot A_{12}}{B_3 \cdot B_{12}} \cdot \frac{B_4 \cdot B_{12}}{A_4 \cdot A_{12}} \cdot \frac{B_1 \cdot B_{23}}{A_1 \cdot A_{23}} \cdot \frac{A_4 \cdot A_{23}}{B_4 \cdot B_{23}} \frac{z'_1}{F_1} \stackrel{def}{\equiv} \delta_3 \frac{z'_1}{z_1}$$

Denote z'_1/z_1 by k then $z'_2/z_2 = \delta_2\, k$, $z'_3/z_3 = \delta_3\, k$ where δ_2 and δ_3 can be derived as above. The actual computations required for δ_2 and δ_3 could be much less than the formula indicate here. This will be discussed in the robustness section.

5. DERIVING MOTION PARAMETERS

Aside from δ_2 and δ_3 derived in the previous section, a virtual object X which is the set of vectors perpendicular to $\boldsymbol{T}$ is introduced in this section to facilitate the analysis. Object X defines either a plane passing through the origin or the whole 3D space if $\boldsymbol{T} \equiv 0$. We will show how X and $\boldsymbol{R}^t X$ could be derived without the knowledge of $\boldsymbol{R}$. These information in turn give $\boldsymbol{T}$ and $\boldsymbol{R}$ directly. The theorems presented are similar to those in T-H method, because they both employ similar form of a key equation. Although the theorems are similar, the nature of the computations are quite different - one is well-conditioned and the other (T-H method) may be ill-conditioned.

From the previous section, the equations governing the two frames can be written as:

$$\boldsymbol{R} z_1 A_1 = k z_1 B_1 - \boldsymbol{T} \tag{5.1}$$

$$\boldsymbol{R} z_2 A_2 = k \delta_2 z_2 B_2 - \boldsymbol{T} \tag{5.2}$$

$$\boldsymbol{R} z_3 A_3 = k \delta_3 z_3 B_3 - \boldsymbol{T} \tag{5.3}$$

Taking the scalar products of both sides of equations (5.1-5.3) with X (the virtual object), we obtain

$$X \cdot \boldsymbol{R} z_1 A_1 = X \cdot k z_1 B_1$$
$$X \cdot \boldsymbol{R} z_2 A_2 = X \cdot \delta_2 z_2 B_2$$
$$X \cdot \boldsymbol{R} z_3 A_3 = X \cdot k \delta_3 z_3 B_3$$

Factoring z_i out, we get:

$$\boldsymbol{R}^t X \cdot A_1 = kX \cdot B_1$$
$$\boldsymbol{R}^t X \cdot A_2 = kX \cdot \delta_2 B_2$$
$$\boldsymbol{R}^t X \cdot A_3 = kX \cdot \delta_3 B_3$$

Let $Y = \boldsymbol{R}^t X$ and rewrite the above equations in vector form:

$$\begin{bmatrix} A_1 & A_2 & A_3 \end{bmatrix}^t Y = k \begin{bmatrix} B_1 & \delta_2 B_2 & \delta_3 B_3 \end{bmatrix}^t X$$

Hence

$$Y = k \left([B_1 \;\; \delta_2 B_2 \;\; \delta_3 B_3] \, [A_1 \, A_2 \, A_3]^{-1} \right)^t X = kAX \tag{5.4}$$

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

Since $Y = \boldsymbol{R}^t X$, we know $Y^t Y = X^t X$. From (5.4), we deduce

$$k^2 X^t A^t A X = X^t X \tag{5.5}$$

Note that k is unknown and the conditions for A to be nonsingular require that A_1, A_2, A_3 and B_1, B_2, B_3 to be nocollinear, respectively. It is interesting to note that the similarity and the difference between our key equation (5.5) and the key equation (16) in T-H method. The similarity is that they both have the same form. The

difference is that the points lying on actual planar patch satisfy equation (16) in T-H method while points on the virtual planar patch satisfy (5.5) above. Now we will rewrite (5.5) into the following system of equations:

$$k^2 X^t A^t A X = c^2 \tag{5.6}$$

$$X^t X = c^2 \tag{5.7}$$

$$X \cdot \boldsymbol{T} = 0 \tag{5.8}$$

where c is an arbitrary constant. Since A^tA is positive definite, its eigenvalues, λ_1, λ_2, λ_3 are all positive and its corresponding eigenvectors, v_1, v_2 and v_3 can be chosen to be orthonomal. Let $Q = [v_1\ v_2\ v_3]^t$. Then (5.6) can be rewritten as

$$k^2 (Q X)^t \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} (Q X) = c^2 \tag{5.9}$$

The solution for equation (5.7) is a sphere with radius c depicted in Figure 2. Taking equation (5.8) into account, the solution of equation (5.7)-(5.8) is a circle, $\boldsymbol{C}$, on the sphere with $\boldsymbol{T}$ to be its orientation, and with center at the origin of the coordinate system. Note that $\boldsymbol{T}$ is unknown, the position of $\boldsymbol{C}$ is unknown. Since the points of $\boldsymbol{C}$ also satisfy (5.9), $\boldsymbol{C}$ lies on the ellipsoid defined by equation (5.9). As we shall see presently, there are one or two ways to orient $\boldsymbol{C}$. In geometrical terms, there are only one or two ways to cut $\boldsymbol{C}$ out of the ellipsoid when the center of $\boldsymbol{C}$ and the center of the ellipsoid coincide. There are three cases: the ellipsoid has three axes equally long, two axes equally long, and all three axes with different lengths.

Case (i): $\lambda_1 = \lambda_2 = \lambda_3 = \lambda$

This case will make equation (5.9) into $k_2\, \lambda\, X^t X = c^2$ which defines a sphere with radius $c/k\sqrt{\lambda}$. For $\boldsymbol{C}$ to lie on this sphere, k must be equal to $1/\sqrt{\lambda}$. Geometrically,

the two spheres defined by equations (5.7) and (5.9) coincide and $\boldsymbol{C}$ can be oriented in any direction. The following algebraic manipulation will show $\boldsymbol{T} = 0$.

Since $A^t A = \lambda I$, we infer that $A/\sqrt{\lambda}$ is a rotation [2]. From equation (5.5), we know that $\boldsymbol{R}^t X = (A/\sqrt{\lambda})X$. This means that the effect of both rotations $\boldsymbol{R}^t$ and $A/\sqrt{\lambda}$ are the same when X has at least two degrees of freedom. Thus $\boldsymbol{R}^t = A/\sqrt{\lambda}$, and

$$\boldsymbol{R}\ \left[A_1\ A_2\ A_3\right] = k\left[B_1\ \delta_2 B_2\ \delta_3 B_3\right]$$

Expanding the above and comparing to equations (5.1)-(5.3), we deduce that $\boldsymbol{T} = 0$. Theorem 1 summarizes this case.

Theorem 1:

If the eigenvalues of $A^t A$ are all equal and are denoted by λ, then $\boldsymbol{T} = 0$ and

$$\boldsymbol{R} = \frac{1}{\sqrt{\lambda}}\ \left[B_1\ \delta_2 B_2\ \delta_3 B_3\right]\ \left[A_1 A_2 A_3\right]^{-1}$$

Case (ii): $\lambda_1 = \lambda_2 \neq \lambda_3$

The ellipsoid defined by (5.9) has two equally long axes (i.e an ellipsoid of revolution) as depicted in Figure 3. In this case, there is only one way to cut a circle out of the elliposid. In fact, the only way to place $\boldsymbol{C}$ on the ellipsoid with the center of $\boldsymbol{C}$ coinciding with the center of the ellipsoid is to have $\boldsymbol{C}$ lie on the plane defined by v_1 and v_2 and to have v_3 as its orientation. For an algebraic proof, see [8]. Thus $\boldsymbol{T} = v_3$ and $k = 1/\sqrt{\lambda}$. If $y_1 = kAv_1$ and $y_2 = kAv_2$, then $\boldsymbol{R}y_1 = v_1$ and $\boldsymbol{R}y_2 = v_2$. Clearly $\boldsymbol{R}(y_1 \times y_2) = v_1 \times v_2$, and we have theorem 2:

Theorem 2:

If the three eigenvalues $\lambda_1, \lambda_2, \lambda_3$ 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

$$T = v_3 \text{ and } R = [v_1, v_2, v_1 \times v_2]\,[y_1, y_2, y_3]^{-1}$$

where $y_1 = kAv_1$, $y_2 = kAv_1$, $y_3 = kAv_1 \times kAv_2$, and $k = 1/\sqrt{\lambda_2}$.

Case (iii): $\lambda_1 > \lambda_2 > \lambda_3$

Without loss of generality, we will assume that $\lambda_1 > \lambda_2 > \lambda_3$. For this case, equation (5.9) defines an ellipsoid with three different lengths of axes depicted in Figure 4 and there are only two ways to cut the circle out of this ellipsoid. This fact is also mentioned in T-H method. Theorem 3 summaries the result and the proof can be seen in [8]

Theorem 3:

If the three eigenvalues λ_1, λ_2 and 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 $T = v_1 \pm \varepsilon\, v_3$; $R = [u_1\ u_2\ u_1 \times u_2]\,[y_1, y_2\ y_1 \times y_2]^{-1}$ where $u_1 = v_2$, $u_2 = \varepsilon\, v_1 - v_3$ or $u_2 = \varepsilon\, v_1 + v_3$ (decided by T), $y_1 = \frac{1}{\sqrt{\lambda_2}} Au_1$, $y_2 = \frac{1}{\sqrt{\lambda_2}} Au_2$, $k = \frac{1}{\sqrt{\lambda_2}}$ and $\varepsilon = \sqrt{\frac{\lambda_2 - \lambda_3}{\lambda_1 - \lambda_2}}$.

6. Discussion on the Robustness of the Method

One of the most important question on motion problem to date is whether a technique is robust or not. In other words, how sensitive the solution is to changes in the image data. In this section, we will analyze the computational aspect of our technique and show it is very reliable. We will also point out reasons why T-H method has difficulty in being a robust algorithm.

Before we investigate these issues, some basis related to the robustness is stated below for the purpose of clarity [6].

> First, we differentiate between an ill-conditioned problem and ill-conditioned computation. All ill-conditioned computation is ususlly the result of using a numerically unstable algorithm. The key point to understand is: If the problem is ill-conditioned, then no amount of effort, trikery, or talent used in the computation can produce accurate answers except by chance.

The computational steps in T-H method proceeds by solving $\mathbf{A}X = \mathbf{b}$ and then computes singular values of a matrix $\mathbf{E}$. Since numerically stable algorithms exist for both steps, the success of T-H method depends on the inherent stability of the solution of the problem with respect to small changes in the image data. Unfortunately, both matrix eigenvalue problem and linear system of equations may also be ill-conditioned. Depending on the problem instance, T-H computing procedure thus may or may not be robust. Most of the examples reported, though, shows it is quite sensitive to the changes in the image data.

Let us now turn to the proposed technique (requiring five steps) and analyze its robustness as below.

(a) Computing the ratios: δ_2 and δ_3

Formula (6.1) and (6.2) are used to determine δ_2 and δ_3.

$$\delta_2 = \frac{A_2 \cdot A_{13}}{B_2 \cdot B_{13}} \cdot \frac{B_4 \cdot B_{13}}{A_4 \cdot A_{13}} \cdot \frac{B_1 \cdot B_{23}}{A_1 \cdot A_{23}} \cdot \frac{A_4 \cdot A_{23}}{B_4 \cdot B_{23}} \tag{6.1}$$

$$\delta_3 = \frac{A_3 \cdot A_{12}}{B_3 \cdot B_{12}} \cdot \frac{B_4 \cdot B_{12}}{A_4 \cdot A_{12}} \cdot \frac{B_1 \cdot B_{23}}{A_1 \cdot A_{23}} \cdot \frac{A_4 \cdot A_{23}}{B_4 \cdot B_{23}} \tag{6.2}$$

Analysis of $A_k \cdot A_{ij}$ and $B_k \cdot B_{ij}$ are similar, so are δ_2 and δ_3. Hence we will only deal with $A_k \cdot A_{ij}$ and δ_2 for simplicity. Recall that A_{ij} is $A_i \times A_j$. The term $A_k \cdot A_{ij}$ actually defines the volume of the parallelepiped **P** with A_k, A_i, A_j as adjacent edges. A_k, A_i, A_j are 3D vectors that emanate from the camera origin and end at the image coordinates of features k, i, j. From this, the formula (6.1) could be reduced to

$$\delta_2 = \frac{B_4 \cdot B_{13}}{A_4 \cdot A_{13}} \cdot \frac{A_4 \cdot A_{23}}{B_4 \cdot B_{23}} \tag{6.3}$$

Now consider the pyramid defined by A_k, A_i, A_j and the camera origin. From vector geometry, the volume of the pyramid is one sixth of the volume of the parallepiped **P**. Since there exist corresponding terms in the numerator and denomenator, we will interpret the geometric meaning of each term as the volume of a pyramid. On the other hand, the volume of a pyramid is one third of the height times "area of the base". Furthermore the bases of these various pyramids are all on the image plane with the camera origin as its vertex. It can thus be concluded that the ratio of terms are essentially equal to the ratio of base-areas since the heights are all the same (i.e. the focal length). Now we could further rewrite formula (6.3) as (6.4) if we denote the area of the triangle formed by features i, j, k at frame t as Δ^t_{ijk}.

$$\delta_2 = \frac{\Delta^2_{134}}{\Delta^1_{134}} \cdot \frac{\Delta^1_{234}}{\Delta^2_{234}} \tag{6.4}$$

The problem of reliablity of the computations now reduces to how sensitive the area of a triangle is to noise in the vertices (features). Instead of pursuing a rigorous mathematical error estimates, requiring definitions of many terms, we will give a rough estimate. Assuming the noise on average will cause ε difference in the length of a side of a triangle. In other words, the width w of the triangle is purturbed into $w+\varepsilon$ and the height h of the triangle is purturbed into $h+\varepsilon$. Then the error resulted in the computation of δ_2 is

$$\sum_{i=1}^{i=4} \varepsilon_i (w_i+h_i)/w_i h_i$$

where i corresponds to different triangle.

This term is roughly proportional to the ratio of the side and the area of a triangle. For most of the triangles, the constant is very small.

(b) Computing $\Omega = [A_1\ A_2\ A_3]^t[A_1\ A_2\ A_3]$

This step is a straightforward multiplication of two matrices. The computational behavior is well known and quite robust.

(c) Computing Ω^{-1}

The error in computation of an inverse of a matrix depends on the condition of the matrix. But Ω is a symmetric positive definite matrix, thus the problem is well-conditioned.

(d) Computing $A^tA = [B_1\ \delta_2 B_2\ \delta_3 B_3]\Omega^{-1}[B_1\ \delta_2 B_2\ \delta_3 B_3]^t$

This step is a straightforward multiplication of three matrices. The computational behavior is well known again and quite robust.

(e) Computing the eigenvalues and eigenvectors of A^tA

As mentioned above matrix eigenvalue problems may be ill-conditioned. It is however a striking and important fact that for symmetric matrices this cannot happen [7]. In other words, the eigenvalues of a symmetric matrix are insensitive to small changes in the matrix. While eigenvalue problem for symmetric matrix are well-conditioned , the same is not true of the eigenvectors. However A^tA is not only symmetric but also positive definite, and the problem of computing eigenvectors is also well-conditioned [7].

7. Planar Patch of N points

Although experiments in section 8 show our technique is quite robust in a wide range of motion parameters, we will discuss briefly one extension of our technique to deal with the case when more than four points are available. For this case, T-H method has a straightforward least square approach. In other words, the number of equations increased as the number of features increased. In our approach, one could use additional triangles (defined by new points) to increase the precision of δ_2 and δ_3.

As stated above in Eq. (5.5), every pair of four points defines a positive definite matrix $A_\alpha^t A_\alpha$ where α ranges over the possible pairs. Given the assumption that the N points lies in the same planar patch, these $A_\alpha^t A_\alpha$ should have the same set of eigenvalues. This could be observed by enlarging eq. (5.7)-(5.9) to include $A_\alpha^t A_\alpha$ of other pairs. It is clear that the ellipsoids defined by $A_\alpha^t A_\alpha$ should all be of the same size and have the same orientation.

A possible extension of our technique to deal with a network of N points lying on the same plane can now be described as follows. Consider all the pairs of four points.

Each pair will yield $A_\alpha^t A_\alpha$ as illustrated in eq. (5.5); and each $A_\alpha^t A_\alpha$ will generate a triplet of eigenvalues. These triplets, then, form a cluster in a general Hough parameter space where a representative can be chosen. To derive the corresponding eigenvectors, one could use a least square approach to solve the following system of equations.

$$A_\alpha^t A_\alpha v_i = \lambda_i v_i \quad i = 1,2,3 \text{ and } \alpha = 1,...O(N^4).$$

7. SIMULATION RESULTS

Before running simulations to investigate the robustness of our algorithm, we briefly discuss the criterion. We will evaluate the relative mean error of the translational vector. Both the mean error and the standard derivation of $\| \hat{\boldsymbol{R}} - \boldsymbol{R} \|$ in l_2 norm will be evaluated where $\hat{\boldsymbol{R}}$ is the estimate of $\boldsymbol{R}$. To see its geometrical meaning, note that the following holds:

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

It says that the angle between $\boldsymbol{R} x$ and $\hat{\boldsymbol{R}} x$ (for every x) cannot exceed $2 \sin^{-1} (\frac{\| \boldsymbol{R} - \hat{\boldsymbol{R}} \|}{2})$. This can be seen in Figure 4 where $\| \boldsymbol{R} - \hat{\boldsymbol{R}} \|$ is the length defined by $\boldsymbol{R} x$ and $\hat{\boldsymbol{R}} x$. In Table (1) through (12), we use the angle between $\boldsymbol{R} x$ and $\hat{\boldsymbol{R}} x$ instead of the numerical norm. The technique used to estimate the norm $\| \boldsymbol{R} - \hat{\boldsymbol{R}} \|$ is based on Gerschgorin's circle theorem ([5],p.304) for eigenvalue and the theorem that "The norm of A is the square root of the largest eigenvalue of $A^T A$" ([5], p.288).

(a) Simulations

All experiments assume the field of view of the camera is 60° and the object size subtends about 20° to 30°. The image is 512×512 pixels. Twelve sets of extensive simulations were performed and the results are listed in the Table 1 through Table 12.

Each table corresponds to a set of motion parameters and various level of noise in the image data. Although these sets of motion parameters do not exhaust all the possible values, they represent a large range of values. The first column of each Table lists the range of noise. For instance, 0-3 means that noise ranging from 0 to 3 pixels is uniformly added to the horizontal and vertical positions of the data independently. Thus the maximum error between the accurate data and the noisy data is $3\times\sqrt{2}\simeq 4$ pixels. The largest noise in the last row is about 12 pixels. In the second column, we list the relative error in terms of translation vector. In the third column, the first entry is the mean error over 100 experiments, the second is its standard deviation. For instance, the result of the first row of Table 1 says: If the true $\boldsymbol{R}$ and the derived $\hat{\boldsymbol{R}}$ are applied to any vector x, then the angle between the resulting two vectors will not exceed 0.2 degrees on average with standard deviation 0.1 degrees.

(b) Real Images

The experiments consist of two sequence of image data. The first sequence shown in Figure 5a-5b is provided by Professor Wohn of Department of Computer and Information Science at the University of Pennsylvania. These images were taken of a model of campus building with camera mounted on a robot arm and the motion of the camera is controlled by a computer. The ground truth data of motion parameters in these two frames consists of no rotation and the equal translational component along x and y axis.

The second sequence shown in Figure 6a-6b is provided by Professor Thomas Huang of Coordinated Science laboratory at the University of Illinois. These two images correspond to the eleventh and the seventeenth frames of a long image sequence. The size of the original image is 4096×4096 with two bytes resolution per

pixel. These images first are reduced to 256×256 pixels with eight bits resolution due to the capacity of the system. This sequence has no ground truth data. Thus several subjects were asked to estimate the ground truth data. A consensus is that the truck is moving from right to left following a curve (perhaps a rough circle); and the rotational angle is small. This means that the rotational axis is perpendicular to the ground.

The first step for any feature-based technique is to extract the features and match them over time. This step has been discussed extensively in an experimental paper [9]. Instead of reimplementing it, we manually extract four features as seen in Figure (5a)-(6b). The program then takes as input the pixel positions of these four points and generate two sets of motion parameters for each sequence. The rotational angle of a set of computed motion parameters corresponding to the first sequence is 3 degrees. The translational vectors says that the camera is moving on the horizontal plane perpendicular to the vertical axis of the image plane and forms a 30 degrees direction with respect to the x-axis. The ground truth data is about 45 degrees on the horizontal plane and no rotational matrix. Consider the second sequence. Our results shows that a rotational axis close (about 10 degrees) to the axis perpendicular to the ground is found and the rotational angle is about 4 degrees. This is quite consistent to the human visual estimation.

8. Concluding Remarks

Two new concepts are introduced to facilitate the analysis: (1) the ratio of depths over time for every featured point in the two scenes and (2) the motion of a virtual plane normal to the translation vector. Interestingly, these two concepts not only sim-

plify the analysis drastically but also provide a robust computing procedure for motion recovery. It is possible that that the ideas developed here could be employed in the technique dealing with general cases.

The most encouraging results of this technique are the demonstrations on extensive simulations and real images. This provides evidences that real-time feature-based motion algorithm can be expected in the future.

References

1. Tsai, R. Y. and T.S. Huang (1982), "Estimating Three-Dimensional Motion parameters of a Rigid Planar Patch II: Singular Value Decomposition", *IEEE Trans. Acoustic, Speech and Signal Processing, Vol. ASSP-30.*

2. H.H. Nagel and B. Neumann, "ON 3-D Reconstruction from Two Perspective Views", *Proc. IJCAI* 81, Vol. II, Aug. 1981.

3. J.K. Aggarwal, "Motion and Time-Varying IMagery- AN Overview", Proc. of IEEE Workshop on Motion: Representation and Analysis, 1986.

4. Bellman R. (1960), *"Introduction to Matrix Analysis"*, McGraw-Hill, New York. p. 25.

5. Strang, G. (1980), *"Linear Algebra and Its Applications"*, Academic Press, New York.

6. Rice, J. (1981), "Matrix Computations and Mathematical Software" McGraw-Hill. p 111.

7. Parlett, B (1980), *"The Symmetric Eigenvalue Problem"*, Prentice-Hall, New Jersey.

8. C. H. Lee (1988), "On Motion of A Planar Patch", Technical Report 757, Department of Computer Science, Purdue University.

Error	Translation	Angle
1	(10%, 5%)	(3.3, 1.6)
2	(14%, 6%)	(4.5, 1.9)
3	(18%, 7%)	(5.7, 2.4)
4	(19%, 9%)	(6.4, 2.9)
5	(22%, 10%)	(7.3, 3.1)
6	(22%, 9%)	(7.6, 3.7)

Table 1 (10,10,10)

Error	Translation	Angle
1	(9%, 4%)	(2.9, 1.4)
2	(14%, 6%)	(4.6, 2.0)
3	(17%, 8%)	(5.7, 2.6)
4	(20%, 8%)	(6.7, 2.6)
5	(21%, 10%)	(7.3, 3.0)
6	(22% 10%)	(7.5, 3.3)

Table 2 (30, 10, 10)

Error	Translation	Angle
1	(10%, 4%)	(3.2, 1.5)
2	(16%, 6%)	(5.2, 2.2)
3	(19%, 7%)	(6.3, 2.5)
4	(19%, 8%)	(6.5, 2.7)
5	(21%, 10%)	(7.3, 3.2)
6	(23%, 10%)	(7.9, 3.3)

Table 3 (50, 10, 10)

Error	Translation	Angle
1	(10%, 4%)	(3.2, 1.5)
2	(13%, 6%)	(4.4, 2.2)
3	(17%, 8%)	(5.8, 2.7)
4	(20%, 8%)	(7.1, 3.0)
5	(20%, 10%)	(7.3, 3.4)
6	(22%, 9%)	(8.0, 3.2)

Table 4 (60,10,10)

Error	Translation	Angle
1	(10%, 5%)	(3.8, 1.7)
2	(14%, 6%)	(4.9, 2.1)
3	(18%, 7%)	(5.9, 2.4)
4	(19%, 9%)	(6.5, 2.8)
5	(22%, 9%)	(7.1, 2.9)
6	(23%, 9%)	(7.6, 2.9)

Table 5 (10,20,10)

Error	Translation	Angle
1	(10%, 4%)	(5.6, 1.9)
2	(15%, 7%)	(6.3, 2.7)
3	(19%, 8%)	(6.9, 2.9)
4	(20%, 9%)	(6.8, 2.9)
5	(21%, 9%)	(7.4, 3.1)
6	(23% 10%)	(7.5, 3.3)

Table 6 (10, 30, 10)

Error	Translation	Angle
1	(8%, 4%)	(6.9, 1.9)
2	(14%, 7%)	(7.5, 3.0)
3	(17%, 8%)	(7.6, 3.0)
4	(22%, 9%)	(7.7, 3.6)
5	(22%, 10%)	(8.3, 4.2)
6	(24%, 11%)	(8.6, 4.0)

Table 7 (10, 40, 10)

Error	Translation	Angle
1	(9%, 4%)	(3.1, 1.4)
2	(15%, 7%)	(5.7, 2.3)
3	(19%, 8%)	(6.2, 2.7)
4	(22%, 9%)	(7.2, 2.9)
5	(22%, 10%)	(7.6, 3.4)
6	(25%, 11%)	(9.4, 3.5)

Table 8 (10,50,10)

Error	Translation	Angle
1	(9%, 5%)	(3.0, 1.5)
2	(15%, 8%)	(4.8, 2.4)
3	(18%, 8%)	(5.8, 2.7)
4	(21%, 9%)	(6.9, 2.9)
5	(23%, 10%)	(7.1, 3.3)
6	(23%, 9%)	(7.8, 3.1)

Table 9 (10,20,20)

Error	Translation	Angle
1	(10%, 5%)	(3.2, 1.6)
2	(15%, 7%)	(4.8, 2.3)
3	(19%, 9%)	(6.3, 2.8)
4	(21%, 9%)	(7.1, 3.1)
5	(23%, 10%)	(7.8, 3.4)
6	(24% 10%)	(8.2, 3.3)

Table 10 (10, 20, 25)

Error	Translation	Angle
1	(10%, 5%)	(3.4, 1.8)
2	(15%, 6%)	(4.8, 2.2)
3	(17%, 8%)	(5.8, 2.5)
4	(20%, 9%)	(6.7, 2.8)
5	(23%, 10%)	(7.8, 3.3)
6	(25%, 9%)	(8.3, 3.2)

Table 11 (10, 20, 30)

Error	Translation	Angle
1	(9%, 5%)	(3.1, 1.6)
2	(15%, 7%)	(5.0, 2.2)
3	(20%, 9%)	(6.5, 2.9)
4	(22%, 9%)	(7.4, 3.2)
5	(23%, 9%)	(8.1, 3.0)
6	(25%, 10%)	(8.7, 3.2)

Table 12 (10,20,40)

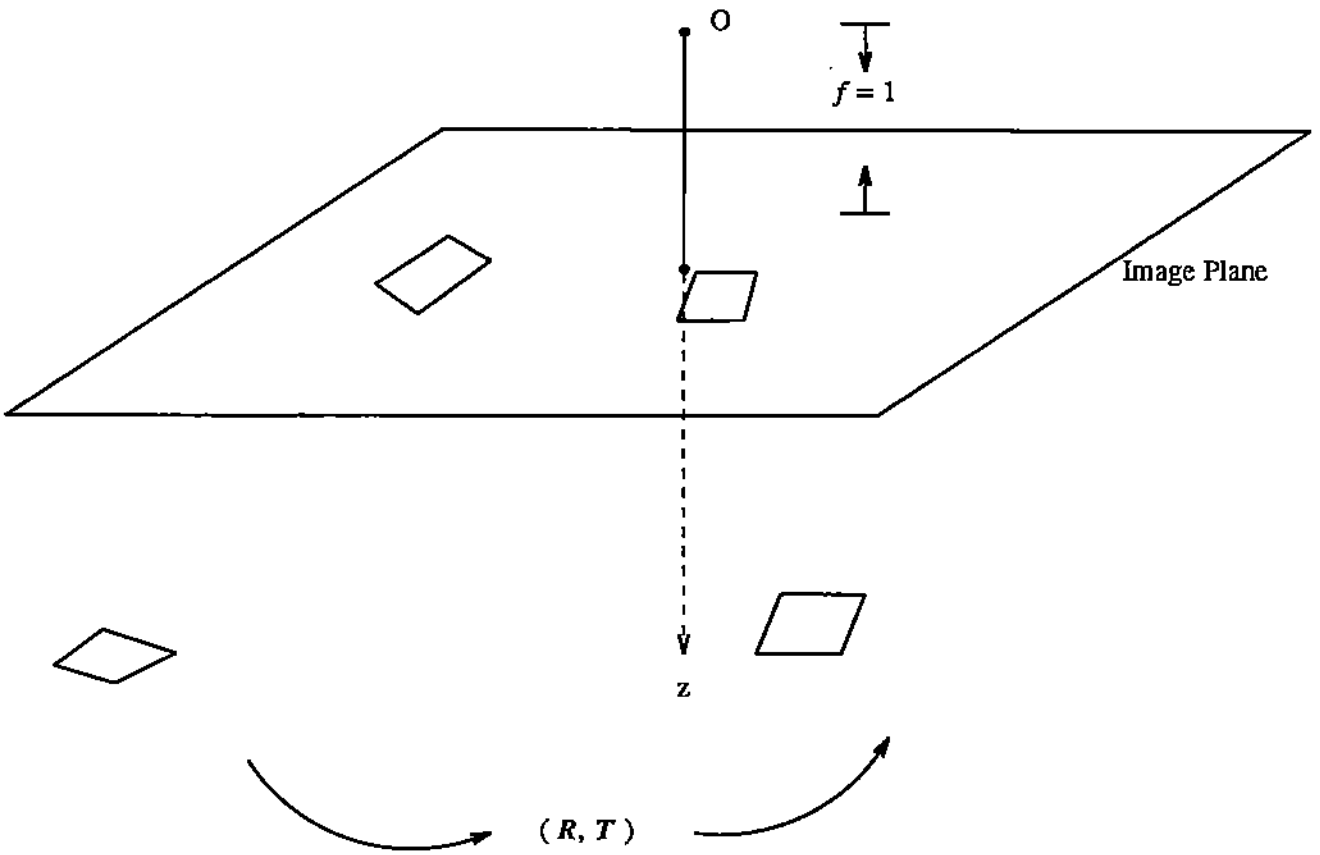

Figure 1. Imaging geometry and the problem.

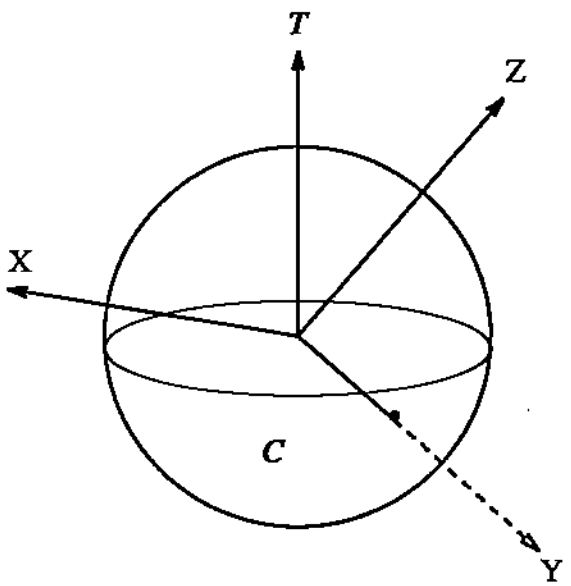

Figure 2: $\boldsymbol{T}$ is the orientation of $\boldsymbol{C}$.

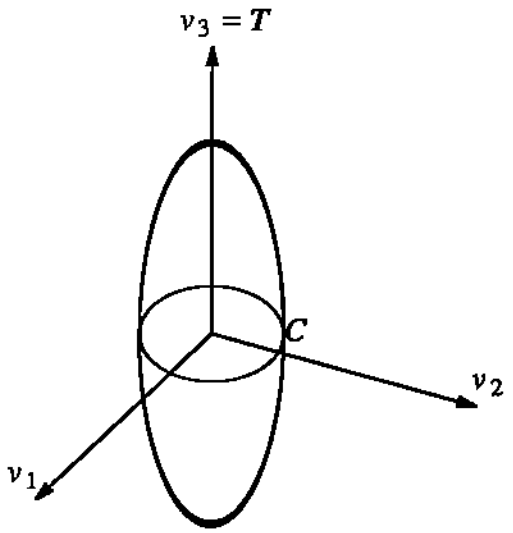

Figure 3: A unique way to orient C (passing through the origin) on the ellipsoid of revolution.

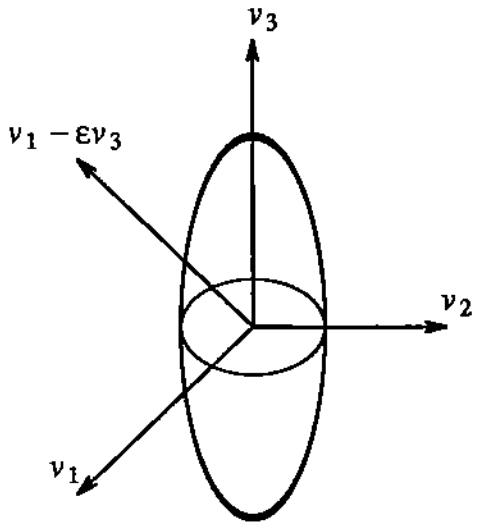

Figure 4: The solid cross section is an ellipse; the dotted cross section represents a circle.

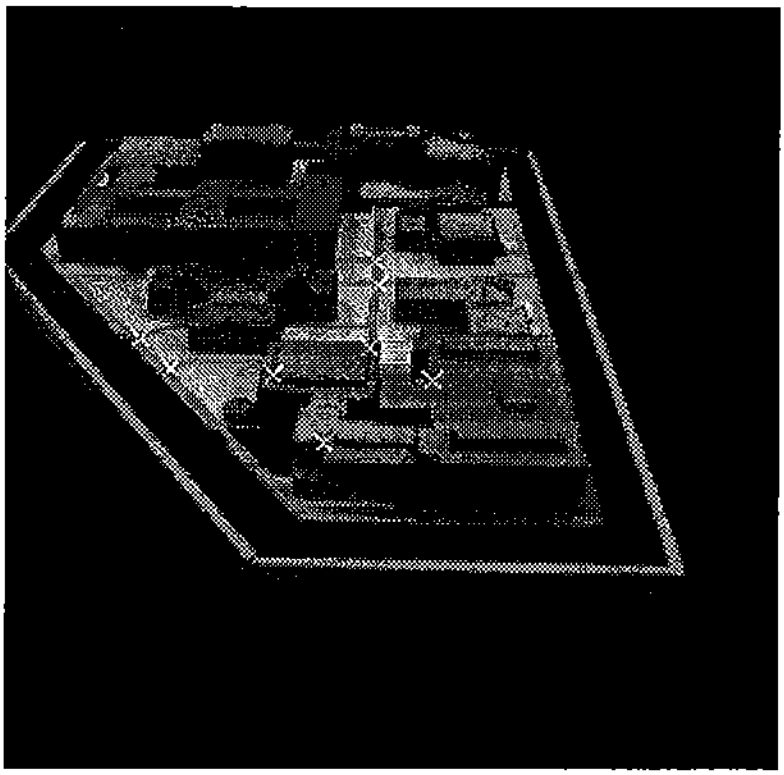

Figure 5(a)

Figure 5(b)

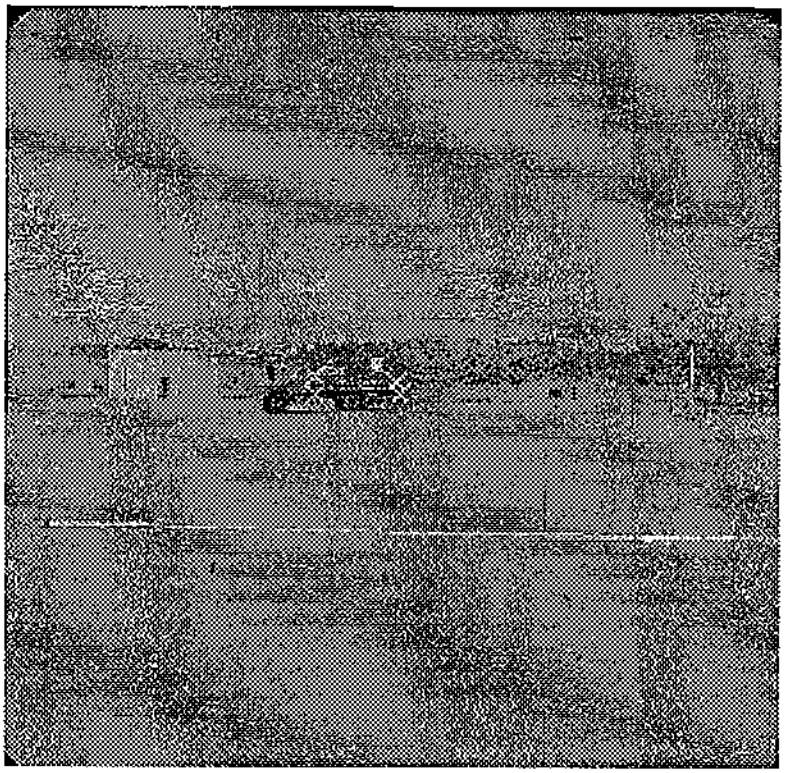

Figure 6(a)

Figure 6(b)