Fast Moment-Based Planar Object Pose Estimation Without Image Correspondence*

Chin-Chun Chang (張欽圳) and Wen-Hsiang Tsai (蔡文祥)†
Department of Computer and Information Science
National Chiao Tung University,
Hsinchu, Taiwan 300, R. O. C.

ABSTRACT

An approach to estimating the 3D pose parameters of a planar object using the perspective projection without feature correspondences is proposed. Two problems are discussed. The first is the estimation of the pose of a planar object with known shape, and the second is the estimation of the relative motion parameters between two views of a planar object with unknown shape and moving without changing the direction of the plane normal. The two problems are treated by an identical mathematical formulation because they both aim to find six motion parameters. Because of the nonlinearity of the perspective projection, the 3D relationship between the two views of a moving planar object is difficult to formulate except when their 2D perspective projections are just different in orientation, position, or scale. This condition can be achieved by iteratively adjusting the orientation of the camera. By comparing the moment invariants of the two views, one way to adjust the orientation of the camera is proposed, in which only two parameters are involved in the iteration process, and the remaining four parameters can be solved analytically. To speed up the iteration part of the proposed method, some improvements on implementation are also proposed. By testing against synthesized data and real images, the experimental results show the feasibiliy of the proposed approach.

Key words: planar patch, pose estimation, relative motion, numerical derivative, ground plane constraint, moment invariants.

I. Introduction

Estimating the 3D pose of a planar object from its 2D perspective projection has received a lot of attention because of its broad applications. Two problems are studied in this paper. The first is the so-called pose estimation problem which aims to estimate the 3D pose of a planar object with known shape from its perspective projection. Two applications of this problem are object positioning, and camera calibration. The second problem is to estimate the relative motion parameters between two views of a moving planar object with unknown shape under the constraint that the plane normals of the planar object in the two views are identical. A straightforward application of the second problem is to estimate the motion parameters of a planar object moving on a plane. In addition, the structure of the planar object can be constructed if the two views are at least related by a translation. This makes it possible to automatically construct the model of a planar object without requiring that the normal of the planar object must be parallel to the optical axis of the camera.

Most existing methods proposed for estimating the pose parameters of a planar object need optical flow or feature correspondences. On the other hand, instead of using the perspective projection, some approximate perspective projection transformations like, orthographic projection and scaled-orthographic projection, have been used to obtain approximate 3D pose parameters. Related researches can be found in [1–8]. However, these methods in general have some difficulty for implementation. For example, feature correspondences are not always easily obtainable, optical flow is also hard to obtain when the separation between two views is large, and the pose parameters estimated from the approximate perspective projection transformations may be poor when the planar object is not far from the camera.

In this paper, an iterative method is proposed to solve the two problems mentioned above under the perspective projection without feature correspondences. In the first problem, since the structure of the planar object is known, it is equivalent to translating this problem into estimating the relative motion parameters between two views of a moving planar object with the knowledge of the 3D pose of the planar object in the first view. Thus, the two problems can both be formulated as the estimation of the relative motion between two views of a moving planar object and can be treated by a single mathematical framework. The main idea of the proposed method is that the relationship between two views of a planar object is easy to formulate from their perspective projections which are different in orientation, position, or scale. Thus, by applying successive camera rotation transformations [9], which are com-

*This work was supported by National Science Council under Grant NSC-86-2213-E-009-114.

†To whom all correspondences should be addressed. E-mail: whtsai@cis.nctu.edu.tw

puted from the gradients of the 2D moments of the two views, to the perspective projections of the planar object observed at the two time instances until the transformed shapes can be described by a rigid motion, the desired relative motion parameters can be recovered from the relationship between the two transformed shapes and the applied camera rotation transformations. Since the proposed method does not need optical flow or feature correspondences, no complicated technique for image processing and feature detection is needed. In addition, because of using the perspective projection model, the proposed method can deal with the case that the planar object is close to the camera.

In the following sections, the framework of the proposed method will be introduced in Section 2. In Section 3, some issues about how to effectively implement the iteration part of the proposed algorithm are discussed. In Section 4, experimental results of the proposed method tested against simulated data and real images are given. In the last section are some concluding remarks.

II. Problem Formulation and Transformation

A planar object $\mathbf{\Omega}$ is a flat object consisting of finite closed regions and lying on a plane called the *supporting plane* of the planar object. Suppose that, initially, the supporting plane of the planar object is the x-y plane of the camera coordinate system, and the centroid of the object is at the origin of the camera coordinate system. Then, the pose of the planar object in a certain view with respect to the camera can be described by a rigid motion which consists of a counterclockwise rotation of the planar object around the z-axis, y-axis, and x-axis of the camera coordinate system by angles γ, β, and α, and then a translation of the planar object in such a way that the centroid of the planar object is located at a 3D point $\mathbf{t}$. Accordingly, the relationship between a point $\mathbf{x}$ on the planar object before and after the rigid motion can be described by

$$\mathbf{y} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z\mathbf{x} + \mathbf{t}, \qquad (1)$$

where

$$\mathbf{R}_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{bmatrix}, \mathbf{R}_y = \begin{bmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{bmatrix},$$

$$\mathbf{R}_z = \begin{bmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} x \\ y \\ 0 \end{bmatrix},$$

$\mathbf{y}$ represents the 3D point of $\mathbf{x}$ after the rigid motion, and the z-component of $\mathbf{t}$ is greater than the focal length f of the camera. In addition, the third column vector of $\mathbf{R}_x\mathbf{R}_y\mathbf{R}_z$ is a unit normal vector of the supporting plane in the camera coordinate system. If the normal of the supporting plane is parallel to the z-axis, we say that the planar object is at a *standard pose*. Let the poses of the planar object in the first view and in the second view be described by Eq. (1), and Eq. (2) below, respectively:

$$\mathbf{y}' = \mathbf{R}_x'\mathbf{R}_y'\mathbf{R}_z'\mathbf{x} + \mathbf{t}'. \qquad (2)$$

The aim of this study is to find the relationship between the two views.

A. *Relationship between two perspective projections of a planar object*

By using the homogeneous coordinate system, the perspective projections of the planar object $\mathbf{\Omega}$ on the image plane in the two views can be described by two 2D shapes,

$$\begin{aligned} \mathbf{S} &= \left\{ \begin{bmatrix} u \\ v \end{bmatrix} \middle| [u \ v \ f \ 1]^t = \mathbf{P} \begin{bmatrix} \mathbf{y} \\ 1 \end{bmatrix} \right. \\ &\left. = \mathbf{P} \begin{bmatrix} \mathbf{I} & \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \right\} \end{aligned}$$

and

$$\begin{aligned} \mathbf{S}' &= \left\{ \begin{bmatrix} u \\ v \end{bmatrix} \middle| [u \ v \ f \ 1]^t = \mathbf{P} \begin{bmatrix} \mathbf{y}' \\ 1 \end{bmatrix} \right. \\ &\left. = \mathbf{P} \begin{bmatrix} \mathbf{I} & \mathbf{t}' \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_x'\mathbf{R}_y'\mathbf{R}_z' & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix} \right\}, \end{aligned}$$

where $\mathbf{S}$ represents the 2D shape of the planar object projected on the image plane in the first view, $\mathbf{S}'$ represents that in the second view, $\mathbf{P}$ is the perspective transformation matrix:

$$\mathbf{P} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & f^{-1} & 0 \end{bmatrix},$$

and $\mathbf{x}$ is a point on the planar object $\mathbf{\Omega}$. In addition, a camera rotation transformation $\mathbf{R}$ on a 2D shape $\mathbf{H}$ is defined [9] by

$$[u \ v \ f \ 1]^t = \mathbf{P} \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \left[u' \ v' \ f \ 1\right]^t, \begin{bmatrix} u' \\ v' \end{bmatrix} \in \mathbf{H},$$

where $\mathbf{R}$ represents a 3D rotation matrix. Furthermore, if $\mathbf{H}$ is the perspective projection of $\mathbf{\Omega}$ in orientation $\mathbf{R}'$ and at position $\mathbf{t}$, then the camera rotation transformation $\mathbf{R}$ of the 2D shape $\mathbf{H}$ satifies the following equality

$$[u \ v \ f \ 1]^t = \mathbf{P} \begin{bmatrix} \mathbf{I} & \mathbf{Rt} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{RR}' & 0 \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix}, \mathbf{x} \in \mathbf{\Omega}.$$

Because of the nonlinearity of the perspective projection model, the transformation from $\mathbf{S}$ to $\mathbf{S}'$ is hard to formulate except when $\mathbf{S}$ and $\mathbf{S}'$ are only different in orientation, position, or scale. Such a circumstance can occur, for example, when $\mathbf{S}$ and $\mathbf{S}'$ are observed with their supporting planes being parallel to the image plane. In other words, there always exist camera rotation transformations $\mathbf{R}_1$ and

$\mathbf{R}_1^{'}$ (e.g., $\mathbf{R}_1 = \mathbf{R}_y^t \mathbf{R}_x^t$, $\mathbf{R}_1^{'} = \mathbf{R}_y^{'t} \mathbf{R}_x^{'t}$) transforming $\mathbf{S}$ and $\mathbf{S}^{'}$, respectively, to $\mathbf{S}_1$ and $\mathbf{S}_1^{'}$such that

$$\begin{bmatrix} u^{'} \\ v^{'} \end{bmatrix} = s \begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}, \quad (3)$$

where $s > 0$ with

$$[u \;\; v \;\; f \;\; 1]^t = \mathbf{P} \begin{bmatrix} \mathbf{I} & \mathbf{R}_1 \mathbf{t} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1 \mathbf{R}_x \mathbf{R}_y \mathbf{R}_z & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix},$$

$$\left[u^{'} \;\; v^{'} \;\; f \;\; 1\right]^t = \mathbf{P} \begin{bmatrix} \mathbf{I} & \mathbf{R}_1^{'} \mathbf{t}^{'} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_1^{'} \mathbf{R}_x^{'} \mathbf{R}_y^{'} \mathbf{R}_z^{'} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ 1 \end{bmatrix},$$

where $[\; u \quad v \;]^t \in \mathbf{S}_1$, and $[\; u^{'} \quad v^{'} \;]^t \in \mathbf{S}_1^{'}$, and $\mathbf{x} \in \boldsymbol{\Omega}$. In this study, we do not consider the "symmetric" shape which may have the same perspective projection at different poses. For convenience, let $\mathbf{y}_1 = \mathbf{R}_1 \mathbf{y}$ and $\mathbf{y}_1^{'} = \mathbf{R}_1^{'} \mathbf{y}^{'}$. Also, from Eq. (1) and Eq. (2), the relationship between $\mathbf{y}_1$ and $\mathbf{y}_1^{'}$ can be described by a rigid transformation

$$\mathbf{y}_1^{'} = \overline{\mathbf{R}} \mathbf{y}_1 + \overline{\mathbf{t}} \quad (4)$$

where $\overline{\mathbf{R}}$ is a 3×3 rotation matrix and $\overline{\mathbf{t}}$ is a translation vector. Multiplying $\mathbf{R}_1^{'t}$ to the both sides of Eq. (4) and substituting $\mathbf{R}_1 \mathbf{y}$ and $\mathbf{R}_1^{'} \mathbf{y}^{'}$ for $\mathbf{y}_1$ and $\mathbf{y}_1^{'}$, respectively, we have

$$\mathbf{y}^{'} = \mathbf{R}_1^{'t} \overline{\mathbf{R}} \mathbf{R}_1 \mathbf{y} + \mathbf{R}_1^{'t} \overline{\mathbf{t}} \quad (5)$$

which is the key equation of this study.

Now, we have to know the 3D relationship between two perspective projections of a planar object when they are only different in orientation, position, or scale. Let the perspective projections of a point $\mathbf{y}_1$ on the planar object in the first view and its corresponding point $\mathbf{y}_1^{'}$ in the second view be $[\; u \quad v \;]^t$ and $[\; u^{'} \quad v^{'} \;]^t$, respectively. Then, from Eq. (3), the relationship between the two transformed shapes $\mathbf{y}_1$ and $\mathbf{y}_1^{'}$ in the camera coordinate system can be described as follows:

$$\begin{bmatrix} u^{'} \\ v^{'} \\ f \end{bmatrix} = s \begin{bmatrix} \cos\psi & -\sin\psi & w_1 \\ \sin\psi & \cos\psi & w_2 \\ w_3 & w_4 & w_5 \end{bmatrix} \begin{bmatrix} u \\ v \\ f \end{bmatrix} + \mathbf{g}. \quad (6)$$

where $\mathbf{g} = [\Delta u - sfw_1 \;\; \Delta v - sfw_2 \;\; f - sfw_5 - suw_3 - svw_4]^t$. Let the depths of $\mathbf{y}_1$ and $\mathbf{y}_1^{'}$, the distances between the origin of the camera coordinate system to $\mathbf{y}_1$ and $\mathbf{y}_1^{'}$, be λ and $\lambda^{'}$, respectively. Then, we have $\mathbf{y}_1 = \frac{\lambda}{f}[\; u \quad v \quad f \;]^t$, $\mathbf{y}_1^{'} = \frac{\lambda^{'}}{f}[\; u^{'} \quad v^{'} \quad f \;]^t$. In addition, according to Eq. (6), the relationship between $\mathbf{y}_1$ and $\mathbf{y}_1^{'}$ can also be described by

$$y_1^{'} = \frac{s\lambda^{'}}{\lambda} \begin{bmatrix} \cos\psi & -\sin\psi & w_1 \\ \sin\psi & \cos\psi & w_2 \\ w_3 & w_4 & w_5 \end{bmatrix} y_1 + \frac{\lambda^{'}}{f} \mathbf{g}. \quad (7)$$

Comparing Eq. (4) with Eq. (7), we have $w_1 = w_2 = w_3 = w_4 = 0$, and $w_5 = 1$ because $\overline{\mathbf{R}}$ is a rotation matrix; in addition,

$$\begin{aligned} \overline{\mathbf{R}} &= \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}, & (8) \\ \overline{\mathbf{t}} &= \frac{\lambda^{'}}{f} [\; \Delta u \quad \Delta v \quad f - sf \;]^t, & (9) \\ \lambda^{'} &= \frac{\lambda}{s}. & (10) \end{aligned}$$

Since $\overline{\mathbf{R}}$ and $\overline{\mathbf{t}}$ must be identical for all of the point pairs on the planar object in the first and the second views, as shown in the following, the poses of the planar object in the two views are either at standard poses or only related by a rotation around the *z-axis*.

- Case 1, $\bar{t} \neq \mathbf{0}$: From Eq. (9), we can know that the all of the points on the planar object in the second view must be at the same depth. In addition, from Eq. (10), the depths of the points on the planar object in the first view are also identical. Therefore, the poses of the object in the two views are both standard poses.

- Case 2, $\bar{t} = \mathbf{0}$: Since $\bar{t} = \mathbf{0}$, from Eq. (9), s must be one. In other words, the depth of a point on the planar object in the first view and that of its corresponding point in the second view are equal. Therefore, the poses of the planar object in the two views are only related by a rotation around the *z-axis*.

Accordingly, once $\mathbf{R}_1$, $\mathbf{R}_1^{'}$, and the relationship between $\mathbf{S}_1$ and $\mathbf{S}_1^{'}$ are found, then the rigid motion between the two views can be identified by Eq. (5). However, for the second problem, λ and $\lambda^{'}$ are unknown, and thus $\overline{\mathbf{t}}$ can only be determined up to a scale factor. For convenience, we assume $\left\|\overline{\mathbf{t}}\right\|_2 = 1$. Before discussing how to obtain $\mathbf{R}_1$, and $\mathbf{R}_1^{'}$ a well known method to compute the parameters s, ψ, Δu, and Δv in Eq. (3) using the moments of $\mathbf{S}_1$ and $\mathbf{S}_1^{'}$ is reviewed as follows.

B. Determining relationship between S_1 and $S_1^{'}$

The $(p+q)$th regular moment of a 2D shape $\mathbf{H}$ is defined as

$$m_{\mathbf{H},pq} = \int\int u^p v^q \mathbf{H}(u,v)\, dudv,$$

where $\mathbf{H}(u,v)$ is a function whose value is one if $[\; u \quad v \;]^t$ is in $\mathbf{H}$, and zero elsewhere. In addition, the $(p+q)$th central moment of the 2D shape $\mathbf{H}$ can be expressed as

$$\eta_{\mathbf{H},pq} = \int\int (u - \overline{u}_{\mathbf{H}})^p (v - \overline{v}_{\mathbf{H}})^q \mathbf{H}(u,v)\, dudv,$$

where $[\; \overline{u}_{\mathbf{H}} \quad \overline{v}_{\mathbf{H}} \;]^t = \left[\; \frac{m_{\mathbf{H},10}}{m_{\mathbf{H},00}} \quad \frac{m_{\mathbf{H},01}}{m_{\mathbf{H},00}} \;\right]^t$ is the centroid of the shape $\mathbf{H}$. First, s can be computed by the areas of the two shapes,

$$s = \sqrt{\frac{m_{\mathbf{S}_1^{'},00}}{m_{\mathbf{S}_1,00}}}. \quad (11)$$

If $\mathbf{S}_1$ and $\mathbf{S}_1^{'}$ are not symmetric shapes, then ψ can be determined by the angles between the principal axes of the two shapes $\mathbf{S}_1$ and $\mathbf{S}_1^{'}$; that is,

$$\psi = \begin{cases} \Delta\theta & \begin{array}{l}\text{if } \eta_{\mathbf{S}_1^{'},30} \text{ is with the same sign as} \\ \left(\eta_{\mathbf{S}_1,30}\cos\Delta\theta - 3\eta_{\mathbf{S}_1,21}\sin\Delta\theta\right)\cos^2\Delta\theta + \\ \left(3\eta_{\mathbf{S}_1,12}\cos\Delta\theta - \eta_{\mathbf{S}_1,03}\sin\Delta\theta\right)\sin^2\Delta\theta;\end{array} \\ \Delta\theta - \pi & \text{otherwise,} \end{cases} \tag{12}$$

where

$$\begin{aligned} \Delta\theta &= \theta_2 - \theta_1, \\ \theta_1 &= \tan^{-1}\frac{\eta_{\mathbf{S}_1,02} - \eta_{\mathbf{S}_1,20} + \sqrt{\left(\eta_{\mathbf{S}_1,02} - \eta_{\mathbf{S}_1,20}\right)^2 + 4\eta_{\mathbf{S}_1,11}^2}}{2\eta_{\mathbf{S}_1,11}}, \\ \theta_2 &= \tan^{-1}\frac{\eta_{\mathbf{S}_1^{'},02} - \eta_{\mathbf{S}_1^{'},20} + \sqrt{\left(\eta_{\mathbf{S}_1^{'},02} - \eta_{\mathbf{S}_1^{'},20}\right)^2 + 4\eta_{\mathbf{S}_1^{'},11}^2}}{2\eta_{\mathbf{S}_1^{'},11}}. \end{aligned}$$

Let the centroids of $\mathbf{S}_1$ and $\mathbf{S}_1^{'}$ be located at $[\ \overline{u} \quad \overline{v}\ ]^t$ and $[\ \overline{u}^{'} \quad \overline{v}^{'}\ ]^t$, respectively. According to Eq. (3), we have

$$\begin{bmatrix} \overline{u}^{'} \\ \overline{v}^{'} \end{bmatrix} = s\begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}\begin{bmatrix} \overline{u} \\ \overline{v} \end{bmatrix} + \begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix}.$$

Hence,

$$\begin{bmatrix} \Delta u \\ \Delta v \end{bmatrix} = \begin{bmatrix} \overline{u}^{'} \\ \overline{v}^{'} \end{bmatrix} - s\begin{bmatrix} \cos\psi & -\sin\psi \\ \sin\psi & \cos\psi \end{bmatrix}\begin{bmatrix} \overline{u} \\ \overline{v} \end{bmatrix}. \tag{13}$$

C. *Determining camera rotation transformations* $\mathbf{R}_1$ *and* $\mathbf{R}_1^{'}$

Now, the remaining question is how to obtain the camera rotation transformations $\mathbf{R}_1$ and $\mathbf{R}_1^{'}$ which make the transformed shapes of $\mathbf{S}$ and $\mathbf{S}^{'}$only different in orientation, position, or scale. In this study, an iterative method is proposed to find $\mathbf{R}_1$ and $\mathbf{R}_1^{'}$ by applying successive camera rotation transformations to $\mathbf{S}$ and $\mathbf{S}^{'}$ until the two transformed shapes $\mathbf{S}_1$ and $\mathbf{S}_1^{'}$ are only different in orientation, position, or scale. A camera rotation transformation can be decomposed to be $\mathbf{R}_z^t\mathbf{R}_y^t\mathbf{R}_x^t$,in which the left-most matrix only affects the orientation and the position of the transformed shape; therefore, the desired camera rotation transformations can be described by the right-most two matrices of $\mathbf{R}_z^t\mathbf{R}_y^t\mathbf{R}_x^t$.

The moment invariants [10] of a 2D shape are independent of orientation, translation, and scaling. Therefore, if the moment invariants of the two transformed shapes $\mathbf{S}_1$ and $\mathbf{S}_1^{'}$ are identical, then the two shapes are only different in orientation, position, or scaling, and thus the desired camera rotation transformations are obtained. Specifically, let $\phi_{\mathbf{H},i}(\alpha,\beta)$ denote the ith moment invariant of the resulting 2D shape after applying the camera rotation transformation described by $\mathbf{R}_y^t\mathbf{R}_x^t$ to the 2D shape $\mathbf{H}$. Now, if we have four parameters α, β, $\alpha^{'}$, and $\beta^{'}$ such that

$$\phi_{\mathbf{S}^{'},i}\left(\alpha^{'},\beta^{'}\right) - \phi_{\mathbf{S},i}(\alpha,\beta) = 0,\ i = 1,\ 2,\ \cdots,\ n, \tag{14}$$

where n is sufficient large, then the desired camera rotation transformations can be obtained accordingly.

However, it is not necessary to search the four parameters for the two problems; only two parameters are enough. In the first problem, the pose of the planar object in the first view is known; in other words, a camera rotation transformation which transforms the planar object to be at a standard pose is known. Thus, we can transform the pose of the planar object in the first view to a standard pose, and only the camera rotation transformation to transform the object in the second view to be at a standard pose need be found. In the second problem, the plane normals of the supporting plane of the planar object in the first and second views are identical. Therefore, if we have a camera rotation transformation to transform the pose of the object in the first view to a standard pose, then we can apply the same camera rotation transformation to the planar object in the second view to be at a standard pose. Let $\mathbf{p}$ denote the vector formed by the parameters to be found and $\Delta\mathbf{p}$ denote the adjustment vector for each iteration. Thus, for each iteration, the update rule is

$$\mathbf{p}_{\text{next iteration}} = \mathbf{p} + \Delta\mathbf{p}.$$

By linearizing Eq. (14), the adjustment vector $\Delta\mathbf{p}$ can be computed as follows.

In the first problem, α and β are known, and the parameters to be found can be represented by $\mathbf{p} = [\ \alpha^{'} \quad \beta^{'}\ ]^t$. Writing Eq. (14) as a first-order approximation, we have

$$\phi_{\mathbf{S}^{'},i}\left(\alpha^{'},\beta^{'}\right) - \phi_{\mathbf{S},i}(\alpha,\beta) + \frac{\phi_{\mathbf{S}^{'},i}\left(\alpha^{'},\beta^{'}\right)}{\partial\alpha^{'}}\Delta\alpha^{'} + \frac{\phi_{\mathbf{S}^{'},i}\left(\alpha^{'},\beta^{'}\right)}{\partial\beta^{'}}\Delta\beta^{'} \cong 0,\ i = 1,\ 2,\ \cdots,\ n.$$

Rewriting the above equation in a matrix form and multiplying weighting factors to them, we can get

$$\mathbf{WJ}\Delta\mathbf{p} = \mathbf{Wc}$$

where

$$\mathbf{J} = \begin{bmatrix} -\frac{\partial\phi_{\mathbf{S}^{'},1}\left(\alpha^{'},\beta^{'}\right)}{\partial\alpha^{'}} & -\frac{\partial\phi_{\mathbf{S}^{'},1}\left(\alpha^{'},\beta^{'}\right)}{\partial\beta^{'}} \\ \vdots & \vdots \\ -\frac{\partial\phi_{\mathbf{S}^{'},n}\left(\alpha^{'},\beta^{'}\right)}{\partial\alpha^{'}} & -\frac{\partial\phi_{\mathbf{S}^{'},n}\left(\alpha^{'},\beta^{'}\right)}{\partial\beta^{'}} \end{bmatrix},$$

$$\mathbf{W} = \operatorname{diag}\left(\left|\phi_{\mathbf{S},1}(\alpha,\beta)\right|, \left|\phi_{\mathbf{S},2}(\alpha,\beta)\right|, \ldots, \left|\phi_{\mathbf{S},n}(\alpha,\beta)\right|\right)^{-1}$$

$$\Delta\mathbf{p} = \begin{bmatrix} \Delta\alpha^{'} \\ \Delta\beta^{'} \end{bmatrix},\ \mathbf{c} = \begin{bmatrix} \phi_{\mathbf{S}^{'},1}\left(\alpha^{'},\beta^{'}\right) - \phi_{\mathbf{S},1}(\alpha,\beta) \\ \vdots \\ \phi_{\mathbf{S}^{'},n}\left(\alpha^{'},\beta^{'}\right) - \phi_{\mathbf{S},n}(\alpha,\beta) \end{bmatrix},$$

in which diag$(d_1, d_2, ..., d_n)$ represents an $n \times n$ diagonal matrix with diagonal elements $d_1, d_2, ..., d_n$. Hence, $\Delta\mathbf{p}$ can be computed by

$$\Delta\mathbf{p} = \left(\mathbf{J}^t\mathbf{W}^t\mathbf{W}\mathbf{J}\right)^{-1}\mathbf{J}^t\mathbf{W}^t\mathbf{W}\mathbf{c}. \tag{15}$$

Since the analytical formulas to compute $\frac{\partial \phi_{\mathbf{S},i}(\alpha,\beta)}{\partial \alpha}$ and $\frac{\partial \phi_{\mathbf{S},i}(\alpha,\beta)}{\partial \beta}$ are too lengthy, we use the numerical derivatives

$$\begin{aligned} \frac{\partial \phi_{\mathbf{S},i}(\alpha,\beta)}{\partial \alpha} &\cong \frac{\phi_{\mathbf{S},i}(\alpha+\epsilon,\beta)-\phi_{\mathbf{S},i}(\alpha,\beta)}{\epsilon}, \\ \frac{\partial \phi_{\mathbf{S},i}(\alpha,\beta)}{\partial \beta} &\cong \frac{\phi_{\mathbf{S},i}(\alpha,\beta+\epsilon)-\phi_{\mathbf{S},i}(\alpha,\beta)}{\epsilon} \end{aligned}$$

instead of the analytical ones in this study where ϵ is a sufficiently small number (0.0001 in this study).

In the second problem, because the desired camera rotation transformations for the two views are the same, we have $\alpha = \alpha'$ and $\beta = \beta'$. Thus, the parameters to be found can be denoted by $\mathbf{p} = [\ \alpha \quad \beta\ ]^t$. Writing Eq. (14) as a first-order approximation, we have

$$\begin{aligned} &\phi_{\mathbf{S}',i}(\alpha,\beta) - \phi_{\mathbf{S},i}(\alpha,\beta) + \frac{\partial \phi_{\mathbf{S}',i}(\alpha,\beta)}{\partial \alpha}\Delta\alpha + \\ &\frac{\partial \phi_{\mathbf{S}',i}(\alpha,\beta)}{\partial \beta}\Delta\beta - \frac{\partial \phi_{\mathbf{S},i}(\alpha,\beta)}{\partial \alpha}\Delta\alpha - \frac{\partial \phi_{\mathbf{S},i}(\alpha,\beta)}{\partial \beta}\Delta\beta \cong 0, \\ &i = 1,\, 2,\, \cdots,\, n. \end{aligned}$$

Rewriting the above equations in a matrix form and multiplying them by some weighting factors, we can get

$$\mathbf{WJ\Delta p} = \mathbf{Wc}$$

where

$$\mathbf{J} = \begin{bmatrix} \frac{\partial \phi_{S,1}(\alpha,\beta)}{\partial \alpha} - \frac{\partial \phi_{S',1}(\alpha,\beta)}{\partial \alpha} & \frac{\partial \phi_{S,1}(\alpha,\beta)}{\partial \beta} - \frac{\partial \phi_{S',1}(\alpha,\beta)}{\partial \beta} \\ \vdots & \vdots \\ \frac{\partial \phi_{S,n}(\alpha,\beta)}{\partial \alpha} - \frac{\partial \phi_{S',n}(\alpha,\beta)}{\partial \alpha} & \frac{\partial \phi_{S,n}(\alpha,\beta)}{\partial \beta} - \frac{\partial \phi_{S',n}(\alpha,\beta)}{\partial \beta} \end{bmatrix},$$

$$\mathbf{\Delta p} = \begin{bmatrix} \Delta\alpha \\ \Delta\beta \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \phi_{\mathbf{S}',1}(\alpha,\beta) - \phi_{\mathbf{S},1}(\alpha,\beta) \\ \vdots \\ \phi_{\mathbf{S}',n}(\alpha,\beta) - \phi_{\mathbf{S},n}(\alpha,\beta) \end{bmatrix},$$

and the weighting factors are

$$\mathbf{W} = \mathrm{diag}\left(\frac{1}{|\phi_{\mathbf{S}',1}(\alpha,\beta)| + |\phi_{\mathbf{S},1}(\alpha,\beta)|}, \ldots, \frac{1}{|\phi_{\mathbf{S}',n}(\alpha,\beta)| + |\phi_{\mathbf{S},n}(\alpha,\beta)|} \right)$$

Accordingly, $\mathbf{\Delta p}$ can be computed by Eq. (15), too.

The iteration process is terminated when $\mathbf{c}^t\mathbf{W}^t\mathbf{Wc}$ is smaller than a threshold value, when the step size of the adjustment vector is small, or when the number of iteration exceeds a threshold value.

If the *a priori* information of the relative motion parameters between the two views is not available, initial guesses for the two parameters must be provided, which are generated by the following steps. First, sample the parameters space (−90°, 90°) × (−90°, 90°) evenly at interval of 30° to generate twenty five parameter pairs. Second, for each parameter pair, apply the camera rotation transformations to the input shapes to generate two transformed shapes. If the parameters are close to the answer, the two transformed shapes will be similar. For simplicity, only up to the second moments are considered in this procedure; therefore, the two transformed shapes can be regarded as two ellipses, and the following difference between the ratios of the lengths of the semimajor and the seminor axes of the two ellipses can be used as a goodness measure for the parameter pair,

$$\left| \frac{\lambda_1}{\lambda_2} - \frac{\lambda_1'}{\lambda_2'} \right|,$$

where λ_1 and λ_2 denote the lengths of the semimajor and the seminor axes of the first ellipse, and λ_1 and λ_2 denote those of the second ellipse. At last, the five best parameter pairs are selected for use as the initial guesses for the above iteration process.

III. Implementation Issues

To make the proposed algorithm more efficient, some improvements in implementation have been taken and described in this section. In this study, a 2D shape is represented by its boundary. This representation is suitable for applying successive camera rotation transformations on the 2D shape. Besides, the moments of the 2D shape can be computed fast from its boundary. To speed up the iterative part of the proposed algorithm, the iteration process is decomposed into two stages. In the first stage, only approximate shapes of the input 2D shapes are created and used in a minimization process until a local minimum is reached. Then, a secondary minimization process is started to find a good solution using the original 2D shapes. At the end of this section, a method to obtain an approximate shape of a 2D shape is proposed, which is based on the moments of the 2D shape.

A. Shape representation

In a digital image, the contour of a closed region can be represented by crack codes [11]. A crack code may have one of the four moving directions: up, down, left, and right. Basically, a crack code corresponds to a boundary segment, but successive crack codes in the same directions can be merged to form a longer boundary segment. A boundary segment is represented by two end points. In addition, the contour of a closed object region is tracked counterclockwise and the contour of a background region is tracked clockwise. Fig. 1 shows an illustrative example. Because a straight line on the image plane after applying a camera rotation transformation remains a straight line, we can apply a camera rotation transformation to the end points of the boundary segments of a 2D shape instead of to all of the points on the boundary segments.

B. Computing moment invariants from boundary segments

In Leu [12], a method to compute the moments of a 2D shape from its boundary segments is introduced. In this method, the moments of a 2D shape can be computed fast by summing up the moments of the triangles formed by the origin of the image plane and the 2D shape's boundary segments. Let $m_{\mathbf{H},ij,pq}$ denote the $(p+q)$th regular moment of the triangle formed by the origin of the image plane and the jth boundary segment of the ith closed contour of a 2D shape **H**. Since the contour of a closed object region is tracked counterclockwise and the contour of a background region is tracked clockwise, the $(p+q)$th regular moment of the 2D shape **H** with c closed regions, each of which consists of n_i boundary segments, $i = 1, 2, ..., c$, can be computed by

$$m_{\mathbf{H},pq} = \sum_{i=1}^{c} \sum_{j=1}^{n_i} m_{\mathbf{H},ij,pq},$$

where the moment expansion of a triangle can be found in Singer [13]. Thus, the central moments of the 2D shape **H** can be computed from its regular moments. Accordingly, the moment invariants $\phi_{\mathbf{H},i}$ of the 2D shape **H** can be obtained.

In this study, the moment invariants based on the second and third moments are used for Eq. (14), and the first seven moment invariants are found enough. To shorten this paper, the formulas for computing the first seven moment invariants of a 2D shape **H** and the zeroth to the third moments of the triangle formed by the origin of the image and a boundary segment are not listed here. Interested readers can find them in [10] and [13].

C. *Polygonal approximations*

The number of boundary segments dominates the speed of the iteration process. To reduce the number of boundary segments, the boundary of a closed region of a 2D shape is approximated in this study with a polygon which is formed by some of the end points of the boundary segments. A simple method to obtain an approximate polygon of a closed region is to successively split the boundary segments of the closed region into two parts until some criterions are satisfied [14]. Since the moments of a 2D shape are used to find the motion parameters, it is proper to define a criterion based on the moments of the 2D shape and its approximate shape. By moving the origin to the centroid of the 2D shape, a criterion adopted in this study for stopping splitting the boundary segments is defined as follows:

$$\frac{|m_{\mathbf{A},pq} - m_{\mathbf{B},pq}|}{|m_{\mathbf{A},pq}|} < \xi, \text{ for } p+q = 0, 1, 2, 3,$$

where **A** represents the region enclosed by the boundary segments and the origin, **B** is the triangle formed by the starting and ending points of the boundary segments and the origin, and ξ is a threshold value. Fig. 2 shows a "B" shape and its approximate polygons with various values of ξ. We can see that the approximate polygons save a dramatic amount of boundary segments. In this study, ξ is taken to be 0.1.

IV. Experimental Results

In this study, the proposed method was implemented on a Pentium-233 PC using the C++ programming language and tested against synthesis data and real images to show its performance. A "B" shape shown in Fig. 2 (a) was used as a model shape for generating test samples at various poses. To analyze noise sensitivity, the test samples were perturbed by some noise. For simplicity, a test sample was generated by perturbing the boundary end points of the perspective projection of the model shape at some pose. The noise was added by the following way

$$\mathbf{u}' = \mathbf{u} + \varepsilon \mathbf{n},$$

where $\mathbf{u}$ is a boundary end point, $\mathbf{u}'$ is the perturbed version of $\mathbf{u}$, $\mathbf{n}$ is a noise vector whose elements are randomly generated from the region $[-1, 1]$, and ε controls the noise level. The relative errors between the estimated pose and the actual pose are defined as follows:

$$\text{relative rotation error} = \frac{\|\text{estimated rotation matrix -actual rotation matrix}\|_F}{\sqrt{3}},$$

$$\text{relative translation error} = \frac{\|\text{estimated translation vector-actual translation vector}\|_2}{\|\text{actual translation vector}\|_2},$$

where $\|\cdot\|_2$ and $\|\cdot\|_F$ represent the 2-norm of a vector, and the Frobenius norm of a matrix [15], respectively. The poses for generating the test samples are listed in Table 1 (a) and (b). The parameters α and β for generating the tested poses are ranged from 0° to 60° because the perspective projection of the tested shape will slant too much to be recognizable when the two parameters are larger than 60°. The experimental results are shown in Figs. 3 and 4. Every point in Fig. 3 and Fig. 4 is the error of an average of 100 test samples. It can be found that orientations and positions both have impact on the stability of the estimated parameters and that it is more difficult to estimate the parameters from a slant view.

Shown in Fig. 5 are some real images used to test the proposed method. This experiment was performed five times to see the repeatability of the proposed method. The results are shown in Table 2 where the variations of the rotation matrix and the translation vector, defined for use in measuring the effectiveness of the pose estimation results, are computed by

$$\text{variation of rotation matrix} = \frac{\|\text{estimated rotation matrix} - \widetilde{R}\|_F}{\sqrt{3}},$$

$$\text{variation of translation vector} = \frac{\|\text{estimated translation vector} - \widetilde{\mathbf{t}}\,\|}{\|\widetilde{\mathbf{t}}\,\|_2}$$

where $\widetilde{\mathbf{R}}$ and $\widetilde{\mathbf{t}}$ denote the means of the estimated rotation matrices and translation vectors, respectively.

From Figs. 3 and 4, and Table 2, we can see that the proposed method is faster in solving the first problem than in solving the second problem although the mathematical formulations of the two problems are similar. This phenomenon can be explained by the fact that the amount of data which the second problem deals with are double of those of the first problem because only one shape is needed to perform the camera rotation transformation in the first problem but two are needed in the second problem.

V. Conclusions

In this study, we have proposed a method to compute the pose parameters of a known planar object and the motion parameters between two views of a unknown planar object moving in such a way that the normal of the supporting plane of the planar object is not changed. The proposed method uses neither feature correspondence nor optical flow; therefore, complicated image processing and feature detection techniques can be avoided. In addition, since the perspective projection model is adopted, the proposed method can deal with the case like that the planar object is not far from the camera which is critical to the method using approximate perspective projection models. To speed up the execution speed of the proposed method, some improvements on the implementation have been proposed. In addition, if the procedures for the camera rotation transformation and moment computation are parallelized, further improvement on speed will be achieved. Both synthesized data and real images are tested in this study. Good experimental results prove the feasibility of the proposed method.

References

[1] R. Y. Tsai and T. S. Huang, "Estimating three-dimensional motion parameters of a rigid planar patch," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. ASSP-29, no. 6, pp. 1147–1152, 1981.

[2] K. I. Kanatani, "Detecting the motion of a planar surface by line and surface integrals," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 13–22, 1985.

[3] K. I. Kanatani, "Tracing planar surface motion from a projection without knowing the correspondence," *Computer Vision, Graphics, and Image Processing*, vol. 29, pp. 1–12, 1985.

[4] M. F. Augusteijn and C. R. Dyer, "Recognition and recovery of the three-dimensional orientation of planar point patterns," *Computer Vision, Graphics, and Image Processing*, vol. 36, pp. 76–99, 1986.

[5] R. Mukundan, "Estimation of quaternion parameters from two dimensional image moments," *CVGIP: Graphical Models and Image Processing*, vol. 54, no. 4, pp. 345–350, 1992.

[6] S. C. Pei and L. G. Liou, "Finding the motion, position and orientation of a planar patch in 3d space from scaled-orthographic projection," *Pattern Recognition*, vol. 27, no. 1, pp. 9–25, 1994.

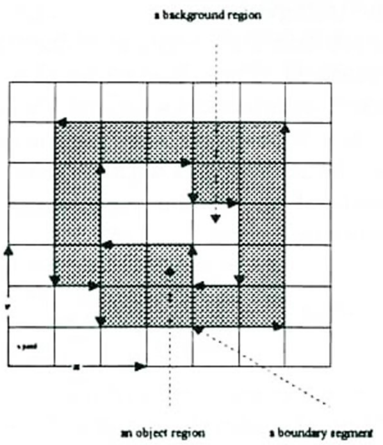

Figure 1 – An illustration of the 2D shape representation.

[7] C. T. D. Lin, D. B. Goldgof, and W. C. Huang, "Motion estimation from scaled orthographic projections without correspondences," *Image and Vision Computing*, vol. 12, no. 2, pp. 95–108, 1994.

[8] R. Mukundan and K. R. Ramakrishnan, "An iterative solution for object pose parameters using image moments," *Pattern Recognition Letters*, vol. 17, pp. 1279–1284, 1996.

[9] K. Kanatani, *Group-Theoretical Methods in Image Understanding*, Spring-Verlag, New York, NY, 1990.

[10] M. K. Hu, "Visual pattern recognition by moment invariants," *IRE Trans. on Information Theory*, vol. 12, pp. 179–187, 1962.

[11] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, vol. 2, Academicc Press, New York, NY, Second edition, 1982.

[12] J. G. Leu, "Computing a shape's moments from its boundary," *Pattern Recognition*, vol. 24, no. 10, pp. 949–957, 1991.

[13] M. H. Singer, "A general approach to moment calculation for polygons and line segments," *Pattern Recognition*, vol. 26, no. 7, pp. 1019–1028, 1993.

[14] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, Reading, 1992.

[15] R. A. Horn and C. R. Johnson, *Matrix Analysis*, Cambridge University Press, New York, NY, 1985.

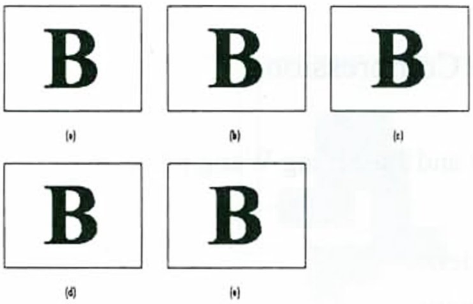

Figure 2 – Polygonal approximations for a 2D shape "B": (a) is the original shape, and (b) to (e) are approximate polygonals with $\xi = 0.05$, 0.1, 0.15, and 0.2, respectively. The numbers of boundary segments used to represent the shapes (a) to (e) are 334, 151, 93, 75, and 64, respectively.

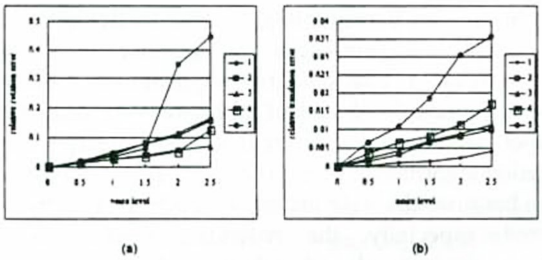

Figure 3 – rotation error, and (b) is for the relative translation error.

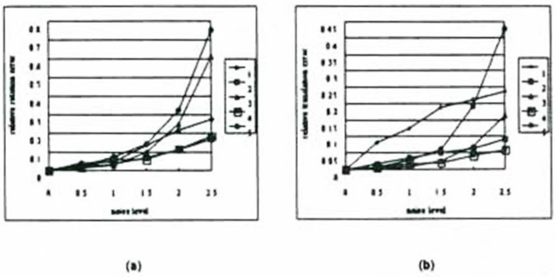

Figure 4 – rotation error, and (b) is for the relative translation error.

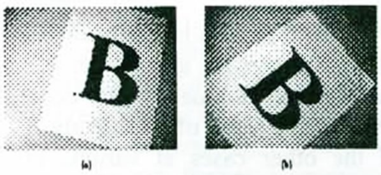

Figure 5 – Real images used in this study: (a) is one view of the test planar shape; (b) is another view of the test planar shape.

Table 1. The poses for generating test samples: (a) is for the first problem, and (b) is for the second problem.

(a)

	orientation	position
no.	(α,β,γ)	(x,y,z)
1	0°,0°,0°	0,0,8000
2	15°,15°,0°	0,0,8000
3	30°,30°,0°	0,0,8000
4	45°,45°,0°	0,0,8000
5	60°,60°,0°	0,0,8000

(b)

	the first view		the second view	
	orientation	position	orientation	position
no.	(α,β,γ)	(x,y,z)	(α,β,γ)	(x,y,z)
1	0°,0°,0°	0,0,7000	0°,0°,0°	100,100,7500
2	15°,15°,0°	0,0,7000	15°,15°,72°	100,100,7500
3	30°,30°,0°	0,0,7000	30°,30°,144°	100,100,7500
4	45°,45°,0°	0,0,7000	45°,45°,216°	100,100,7500
5	60°,60°,0°	0,0,7000	60°,60°,288°	100,100,7500

Table 2. Experimental results for real images: (a) is for the first problem, and (b) is for the second problem.

(a)

	1	2	3	4	5
number of boundary segments	448	452	456	452	452
number of boundary segments of approximate shape	117	123	89	98	98
variation of rotation matrix	0.058	0.052	0.06	0.079	0.035
variation of translation vector	0.062	0.052	0.059	0.081	0.039
computation time (sec)	0.43	0.44	0.38	0.39	0.39

(b)

	1	2	3	4	5
number of boundary segments in the 1st view	448	452	456	452	452
number of boundary segments of approximate shape in the 1st view	117	123	89	98	98
number of boundary segments in the 2nd view	756	750	754	774	764
number of boundary segments of approximate shape in the 2nd view	177	166	157	163	174
variation of rotation matrix	0.011	0.025	0.028	0.069	0.015
variation of translation vector	0.01	0.018	0.032	0.033	0.041
computation time (sec)	0.99	0.93	0.82	0.88	0.88