RECOVERING 3D RIGID MOTIONS WITHOUT CORRESPONDENCE

Tsai-Chia Chou

Center for Automation Research
University of Maryland
College Park, MD 20742, U.S.A.

Ken-ichi Kanatani

Department of Computer Science
Gunma University
Kiryu, Gunma 376, Japan

ABSTRACT

The 3D motion of an object is recovered from its 2D perspective projection images *without using any knowledge of point-to-point correspondence*. The principle is to express the variation of certain image *features* as functionals in terms of motion parameters. Image features may be outstanding points, line segments, or surface patches on the object. Given images of an object before and after motion, together with the depth information of the object before motion, a heuristic estimate of the motion is first computed; and the image before motion is transformed, according to this estimate, so that its position will be close to the other image. Then, the motion that accounts for the remaining small discrepancy is estimated by measuring numerical features on the images. The derivation is based on the optical flow due to infinitesimal motion; and the estimation is done by solving a set of simultaneous *linear* equations in terms of feature values and motion parameters. This process is iterated; after each iteration of estimation, one image is transformed according to the estimated motion so that it is positioned closer and closer to the other image.

1. INTRODUCTION

A major issue in computer vision is to estimate the 3D motion of a rigidly moving object from a sequence of its 2D images. For several years, many researchers have been involved in recovering the 3D structure and motion of an object by defining, computing, and interpreting appropriate information from time varying images. Among these approaches, some consider the problem from the viewpoint of *discrete* sampling of image frames, others from the viewpoint of *continuous* sampling.

From the discrete viewpoint, a finite motion occurs between successive image frames; the resulting displacement between a point in one frame and its corresponding point in the next frame is called the *displacement field*. To estimate the motion, attempts are made to relate motion parameters to displacement fields. Before setting up this relation as a set of simultaneous non-linear equations, the *correspondence problem* must first be solved. Then, numerical methods have to be applied to obtain the motion parameters [18, 20, 21, 24, 25]. However, a difficulty arises because a non-linear system of equations often does not have a *unique* solution; and the noise in the images of corresponding points gives rise to further difficulty in obtaining a consistent solution [1, 24].

From the continuous viewpoint, an infinitesimal motion is observed between successive frames; and the corresponding movement of image points is called *optical flow*. Estimating this kind of motion may be simpler to some extent, and exact analytical solutions can be obtained if the object is a planar surface [16, 17, 19, 23, 27]. There also are methods which do not require correspondence between successive frames [8-13, 26]. However, it is difficult to detect optical flow since the very small image discrepancy may be buried in other sources of errors.

It would be desirable to have a method of recovering 3D motions of finite magnitudes without having to solve the correspondence problem. In [7] an attempt is made to do this by assuming that the projection is orthographic and objects are planar. In [2, 3] two cameras with the same image planes are used so that the depth information, both before and after motion, can be determined. [15] uses a similar principle to that described here, but considers only a single planar object. The method proposed here is different. Suppose that only one camera is used and only the depth information of the initial frame is given a priori, either as in model-based vision or by some other type of preprocessing. In order to compute the motion which transforms the initial image into a target image, we first move the initial image close to the target image by a heuristic estimate. Next, we measure the discrepancy of appropriate numerical *features* between these two images. Then, the estimate of the motion is computed from a set of simultaneous *linear* equations which relates the variation of the image features to the motion parameters. However, this estimate may not be accurate enough. In order to improve the accuracy, an expected image is generated by transforming the initial image according to the estimated motion. Then, the process is repeated, replacing the initial image with the expected image. Using the estimates obtained at each iteration, the expected image will get closer and closer to the target image.

2. IMAGE VARIATION UNDER RIGID MOTIONS

2.1. Geometric Model of the Imaging Process

Perspective projection is used here as the geometric model for the imaging process. Let the XY-plane of a 3D Cartesian coordinate system be the image plane, and the focus be located at the point $(0,0,-f)$, where f is the *focal length*. Thus, a point (X,Y,Z) in the scene is mapped into a point (x,y) on the image plane by

$$x=\frac{fX}{f+Z}, \qquad y=\frac{fY}{f+Z}. \tag{2.1}$$

2.2. Image Variation under Finite Motion

Consider a 3D object which moves rigidly in the scene. Any rigid motion can be described as a translation following a rotation around some *reference point*. Let (a, b, c) be the translation component and $R=(r_{ij})_{3\times3}$ be the rotation component with respect to an arbitrary reference point (X_0,Y_0,Z_0). Thus, a point (X,Y,Z) on a 3D object will be mapped into the point (X',Y',Z') under rigid motion by

$$\begin{bmatrix}X'\\Y'\\Z'\end{bmatrix}=\begin{bmatrix}X_0\\Y_0\\Z_0\end{bmatrix}+\begin{bmatrix}a\\b\\c\end{bmatrix}+R\begin{bmatrix}X-X_0\\Y-Y_0\\Z-Z_0\end{bmatrix}. \tag{2.2}$$

Here, r_{ij}, $i,j=1,2,3$, and a, b, c are called the *motion parameters*.

The variation in the images of a moving object is called the *displacement field*. Let (x,y) and (x',y') be images of (X,Y,Z) and (X',Y',Z'), respectively. Then, when point (X,Y,Z) is moved by the rigid motion (2.2) to point (X',Y',Z'), the displacement field is derived, from (2.1), (2.2) , and $x'=fX'/(f+Z')$, $y'=fY'/(f+Z')$, as follows:

$$x'=f\frac{r_{11}x+r_{12}y+A_1f}{r_{31}x+r_{32}y+A_3f}, \qquad y'=f\frac{r_{21}x+r_{22}y+A_2f}{r_{31}x+r_{32}y+A_3f}, \tag{2.3}$$

where

$$A_1=\frac{1}{f+Z}(a+X_0-r_{11}X_0-r_{12}Y_0-r_{13}(Z_0-Z)),$$

$$A_2=\frac{1}{f+Z}(b+Y_0-r_{21}X_0-r_{22}Y_0-r_{23}(Z_0-Z)), \tag{2.4}$$
$$A_3=\frac{1}{f+Z}(f+c+Z_0-r_{31}X_0-r_{32}Y_0-r_{33}(Z_0-Z)).$$

2.3. Image Variation under Infinitesimal Motions

Consider as a special case an infinitesimal motion. If the rotation angle Ω is close to zero, the rotation is close to the identity. Thus, matrix R will have the form

$$R=I+\Delta R \tag{2.5}$$

where I is the unit matrix and

$$\Delta R=\begin{bmatrix} 0 & -\Omega_3 & \Omega_2 \\ \Omega_3 & 0 & -\Omega_1 \\ -\Omega_2 & \Omega_1 & 0 \end{bmatrix}+\cdots. \tag{2.6}$$

such that $\Omega_1=\Omega n_1$, $\Omega_2=\Omega n_2$, $\Omega_3=\Omega n_3$, and $\cdots$ denotes higher order terms in Ω. This infinitesimal rotation can be regarded as a rotation around an axis whose orientation is $(\Omega_1,\Omega_2,\Omega_3)$ by an angle $\sqrt{\Omega_1^2+\Omega_2^2+\Omega_3^2}$ (rad) counterclockwise. Hence, a, b, c, Ω_1, Ω_2, Ω_3 can be regarded as the motion parameters of an infinitesimal motion.

The displacement field under infinitesimal motion is called *optical flow*. By equations (2.3) and (2.5), the optical flow is

$$x'=x+\Delta x, \quad y'=y+\Delta y \tag{2.7}$$

such that

$$\begin{aligned}\Delta x&=A+Cx-Dy+(Ex+Fy)x+\cdots,\\ \Delta y&=B+Dx+Cy+(Ex+Fy)y+\cdots,\end{aligned} \tag{2.8}$$

where $\cdots$ denotes higher order terms in a, b, c, Ω_1, Ω_2, Ω_3, and

$$\begin{aligned}A&=\frac{f}{f+Z}(a+(Z-Z_0)\Omega_2+Y_0\Omega_3),\\ B&=\frac{f}{f+Z}(b+(Z_0-Z)\Omega_1-X_0\Omega_3),\\ C&=-\frac{1}{f+Z}(c-Y_0\Omega_1+X_0\Omega_2), \quad D=\Omega_3,\\ E&=\Omega_2/f, \quad F=-\Omega_1/f.\end{aligned} \tag{2.9}$$

Parameters A,B,C,D,E,F are referred to as *flow parameters*.

By keeping only terms which are linear in the motion parameters a, b, c, Ω_1, Ω_2, Ω_3, equation (2.8) can be rewritten as

$$\begin{aligned}\Delta x&=u_1a+u_2b+u_3c+u_4\Omega_1+u_5\Omega_2+u_6\Omega_3+\cdots,\\ \Delta y&=v_1a+v_2b+v_3c+v_4\Omega_1+v_5\Omega_2+v_6\Omega_3+\cdots,\end{aligned} \tag{2.10}$$

where $\cdots$ denotes higher order terms in a, b, c, Ω_1, Ω_2, Ω_3, and

$$u_1=\frac{f}{f+Z}, \quad u_2=0, \quad u_3=-\frac{x}{f+Z},$$
$$u_4=\frac{xY_0}{f+Z}-\frac{xy}{f}, \quad u_5=\frac{f(Z-Z_0)-xX_0}{f+Z}+\frac{x^2}{f}, \quad u_6=\frac{fY_0}{f+Z}-y,$$
$$v_1=0, \quad v_2=\frac{f}{f+Z}, \quad v_3=-\frac{y}{f+Z}, \tag{2.11}$$
$$v_4=\frac{f(Z_0-Z)+yY_0}{f+Z}-\frac{y^2}{f}, \quad v_5=\frac{xy}{f}-\frac{yX_0}{f+Z}, \quad v_6=x-\frac{fX_0}{f+Z}.$$

3. RECOVERING MOTION PARAMETERS USING IMAGE FEATURES

3.1. Image Features

An image is a function X defined on the image plane. The value $X(x,y)$ depends on the type of imaging device used. For example, if the imaging device is a monochrome camera, then $X(x,y)$ is the image intensity, or gray level, at point (x,y). If the imaging device is a color camera, $X(x,y)$ may be a vector value with three components corresponding to the intensities of R, G, B, respectively. For an abstract model, $X(x,y)$ may even be defined as taking on delta-function-like values to denote the fact that X consists of only some abstract points or lines. In any case, a *functional* F can be defined over the set of images; i.e., F will map an image X into a real number. For example, F may be defined as the total length of all lines appearing in X. Thus, a *feature* of an image can be defined as a functional on it. This is also called a *property* by Rosenfeld and Kak [22]; and this type of feature was also used in [4, 5, 8].

As an object moves in 3D space, its image varies. Hence, the numerical values of features may also change. If the features are defined appropriately, the variation in these feature values will reveal information about the undergoing motion. Namely, if the variation of the image features can be expressed in terms of the motion parameters only, it is possible to recover the motion parameters by computing the variation of the image features alone. Since the features are defined for the entire image, i.e., globally, no information about the correspondence between consecutive frames is required in computing the variation of the image features.

3.2. Estimating Infinitesimal Motions

By (2.10), the optical flow is linear in terms of the motion parameters. Therefore, the case of infinitesimal motions is considered first.

Let F be a feature defined over a set of images. Let $F[X]$ be a feature value of the image before motion, and $F[X']$ be the value of the same feature after motion. Under an infinitesimal motion, if the feature functional $F[\,.\,]$ is smooth, the value $F[X']$ is also infinitesimally different from $F[X]$, i.e., the difference

$$\Delta F[X]\equiv F[X']-F[X] \tag{3.1}$$

is infinitesimally small. $\Delta F[X]$ can be expanded as a Taylor series

$$\begin{aligned}\Delta F[X]=&F_1[X]a+F_2[X]b+F_3[X]c\\ &+F_4[X]\Omega_1+F_5[X]\Omega_2+F_6[X]\Omega_3+\cdots,\end{aligned} \tag{3.2}$$

where $\cdots$ denotes higher order terms in a, b, c, Ω_1, Ω_2, Ω_3. Note that the *forms* of $F_j[\,.\,]$, $j=1,\ldots,6$, are functionals derived from the given functional $F[\,.\,]$ and are *independent of the image* X. Hence, $F_j[\,.\,]$, $j=1,\ldots,6$, are *known functionals* which can be derived a priori.

Thus, if the difference $\Delta F[X]=F[X']-F[X]$ is observed for two images X and X' before and after the motion respectively, (3.2) gives a *linear constraint* on the motion parameters a, b, c, Ω_1, Ω_2, Ω_3 (neglecting higher order terms). Therefore, if at least six independent feature functionals $F^{(i)}[X]$, $i=1,2,\cdots$, are provided, a set of simultaneous *linear* equations is available to determine the motion parameters a, b, c, Ω_1, Ω_2, Ω_3 in the form

$$KP=D, \tag{3.3}$$

$$K=(K_{ij})=(F_j^{(i)}[X]), \quad i=1,2,\cdots, \quad j=1,\ldots,6,$$

$$P=(\,a\ b\ c\ \Omega_1\ \Omega_2\ \Omega_3\,)^T, \quad D=(\Delta F^{(i)}[X])^T, \quad i=1,2,\cdots$$

When more than six equations are available, the least square method can be applied to solve (3.3).

Note that, by (2.14), in order to compute matrix K, i.e., all the values $F_j^{(i)}[X]$, it is required to know the depth information Z of the object in the first frame X. And since D, the variation of functional F, is computed over the same feature on both X and X', an assumption must be made that only those common features appearing in both frames are taken into account.

4. FEATURES FOR MOTION RECOVERY

4.1. Point Features

If the images of the observed object always contain points, an appropriate definition of a feature is

$$F[X]=\sum_{P_i\in S}m(x_i,y_i). \tag{4.1}$$

Here S is a region in the image plane which contains only the object of interest. P_i is a point located at (x_i,y_i). m is an arbitrarily given *weight function* defined over the region S. This type of feature was also used in [14].

If each feature point (x_i, y_i) is displaced by $(\Delta x_i, \Delta y_i)$, the value $F[X]$ varies by

$$\Delta F[X] = \sum_{P_i \in S} [\frac{\partial m}{\partial x}(x_i, y_i)\Delta x_i + \frac{\partial m}{\partial y}(x_i, y_i)\Delta y_i + \cdots], \tag{4.2}$$

where $\cdots$ denotes higher order terms in Δx, Δy. Substituting the equations of optical flow (2.10), functionals $F_j[X]$, $j=1, \ldots, 6$ are derived as follows:

$$\Delta F[X] = F_1 a + F_2 b + F_3 c + F_4 \Omega_1 + F_5 \Omega_2 + F_6 \Omega_3 + \cdots,$$

$$F_1[X] = \sum_{P_i \in S} \frac{f}{f+Z_i} \frac{\partial m}{\partial x}(x_i, y_i),$$

$$F_2[X] = \sum_{P_i \in S} \frac{f}{f+Z_i} \frac{\partial m}{\partial y}(x_i, y_i),$$

$$F_3[X] = \sum_{P_i \in S} \frac{-1}{f+Z_i} [x_i \frac{\partial m}{\partial x}(x_i, y_i) + y_i \frac{\partial m}{\partial y}(x_i, y_i)], \tag{4.3}$$

$$F_4[X] = \sum_{P_i \in S} [(\frac{x_i Y_0}{f+Z_i} - \frac{x_i y_i}{f}) \frac{\partial m}{\partial x}(x_i, y_i) + (\frac{f(Z_0 - Z_i)}{f+Z_i} + \frac{y_i Y_0}{f+Z_i} - \frac{y_i^2}{f}) \frac{\partial m}{\partial y}(x_i, y_i)],$$

$$F_5[X] = \sum_{P_i \in S} [(\frac{f(Z_i - Z_0)}{f+Z_i} - \frac{x_i X_0}{f+Z_i} + \frac{x_i^2}{f}) \frac{\partial m}{\partial x}(x_i, y_i) - (\frac{y_i X_0}{f+Z_i} + \frac{x_i y_i}{f}) \frac{\partial m}{\partial y}(x_i, y_i)],$$

$$F_6[X] = \sum_{P_i \in S} [(\frac{fY_0}{f+Z_i} - y_i) \frac{\partial m}{\partial x}(x_i, y_i) - (\frac{fX_0}{f+Z_i} - x_i) \frac{\partial m}{\partial y}(x_i, y_i)].$$

Note that Z_i in the above equations is the depth of point P_i. Thus, if at least six independent weight functions m_i, $i=1,2, \cdots$ are applied, at least six linear equations in terms of a, b, c, Ω_1, Ω_2, Ω_3 are obtained. Hence, given the depth information of the observed object in the first frame, the motion parameters can be recovered by the procedure described in the previous section.

4.2. Line Features

If images of the observed object always contain lines, the sum of the line integrals over these lines gives an appropriate definition of a feature:

$$F[X] = \sum_{L_i \in S} \int_{L_i} m(x, y) ds \tag{4.4}$$

Here S is a region in the image plane which contains only the object of interest. L_i is a line segment used as a primitive of the feature. m is an arbitrarily given weight function over the region S. This type of feature was also used in [14].

If point (x, y) on line segment L_i is displaced by $(\Delta x, \Delta y)$, the value $F[X]$ is varied by

$$\Delta F[X] = \sum_{L_i \in S} \int_{L_i} [\frac{\partial m}{\partial x}\Delta x + \frac{\partial m}{\partial y}\Delta y + (n_1^2 \frac{\partial \Delta x}{\partial x} + n_1 n_2 (\frac{\partial \Delta x}{\partial y} + \frac{\partial \Delta y}{\partial x}) + n_2^2 \frac{\partial \Delta y}{\partial y}) m] ds + \cdots, \tag{4.5}$$

where s and (n_1, n_2) are the arc length and unit tangent vector respectively along the line segment L_i, and $\cdots$ denotes higher order terms in Δx, Δy and their derivatives. Substituting the equations of optical flow (2.13), functionals $F_j[X]$, $j=1, \ldots, 6$ are derived as follows:

$$\Delta F[X] = F_1 a + F_2 b + F_3 c + F_4 \Omega_1 + F_5 \Omega_2 + F_6 \Omega_3 + \cdots,$$

$$F_1[X] = \sum_{L_i \in S} \int_{L_i} \frac{f}{f+Z} \frac{\partial m}{\partial x} ds,$$

$$F_2[X] = \sum_{L_i \in S} \int_{L_i} \frac{f}{f+Z} \frac{\partial m}{\partial y} ds,$$

$$F_3[X] = \sum_{L_i \in S} \int_{L_i} \frac{-1}{f+Z} [x \frac{\partial m}{\partial x} + y \frac{\partial m}{\partial y} + m(n_1^2 + n_2^2)] ds, \tag{4.6}$$

$$F_4[X] = \sum_{L_i \in S} \int_{L_i} [(\frac{xY_0}{f+Z} - \frac{xy}{f}) \frac{\partial m}{\partial x} + (\frac{f(Z_0 - Z)}{f+Z} + \frac{yY_0}{f+Z} - \frac{y^2}{f}) \frac{\partial m}{\partial y} + m((\frac{Y_0}{f+Z} - \frac{y}{f}) n_1^2 - \frac{x}{f} n_1 n_2 + (\frac{Y_0}{f+Z} - \frac{2y}{f}) n_2^2)] ds,$$

$$F_5[X] = \sum_{L_i \in S} \int_{L_i} [(\frac{f(Z - Z_0)}{f+Z} - \frac{xX_0}{f+Z} + \frac{x^2}{f}) \frac{\partial m}{\partial x} - (\frac{yX_0}{f+Z} - \frac{xy}{f}) \frac{\partial m}{\partial y} + m((\frac{2x}{f} - \frac{X_0}{f+Z}) n_1^2 + \frac{y}{f} n_1 n_2 + (\frac{x}{f} - \frac{X_0}{f+Z}) n_2^2)] ds,$$

$$F_6[X] = \sum_{L_i \in S} \int_{L_i} [(\frac{fY_0}{f+Z} - y) \frac{\partial m}{\partial x} - (\frac{fX_0}{f+Z} - x) \frac{\partial m}{\partial y}] ds.$$

Note that Z in the above equations is the depth along the 3D line whose image is L_i. Since the depth information of the observed object in the first frame is assumed to be given, Z can be computed as a function of s.

4.3. Region Features

In most cases, the image of a 3D object consists of regions with closed contours. If certain regions can always be identified as the object moves, an appropriate definition of a feature is given as the sum of area integrals over these regions:

$$F[X] = \sum_{A_i \in S} \iint_{A_i} m(x, y) dx dy, \tag{4.7}$$

Here S is the domain in the image plane which contains only the object of interest. A_i is a region identified and used as a primitive. m is an arbitrarily given weight function over S. This type of feature was also used in [9]. Note that an area integral can always be converted into a line integral along its contour by *Green's theorem*. Namely, there exist two functions M, N such that

$$m = \frac{\partial N}{\partial x} - \frac{\partial M}{\partial y}, \tag{4.8}$$

and

$$\iint_{A_i} m(x, y) dx dy = \int_{C_i} [M(x, y) n_1 + N(x, y) n_2] ds, \tag{4.9}$$

where s and (n_1, n_2) are the arc length and unit tangent vector respectively along the contour C_i of region A_i traced counterclockwise.

If point (x, y) on the image plane is displaced by $(\Delta x, \Delta y)$, the value $F[X]$ varies by

$$\Delta F[X] = \sum_{C_i \in S} \int_{C_i} (n_2 \Delta x - n_1 \Delta y) m ds + \cdots, \tag{4.10}$$

$$= \sum_{A_i \in S} \iint_{A_i} [\frac{\partial m}{\partial x}\Delta x + \frac{\partial m}{\partial y}\Delta y + (\frac{\partial \Delta x}{\partial x} + \frac{\partial \Delta y}{\partial y}) m] dx dy + \cdots,$$

where $\cdots$ denotes higher order terms in Δx, Δy and their derivatives. Substituting the equations of optical flow (2.10), functionals $F_j[X]$, $j=1, \ldots, 6$ are derived as follows:

$$\Delta F[X] = F_1 a + F_2 b + F_3 c + F_4 \Omega_1 + F_5 \Omega_2 + F_6 \Omega_3 + \cdots,$$

$$F_1[X] = \sum_{C_i \in S} \int_{C_i} \frac{f}{f+Z} n_2 m ds,$$

$$F_2[X]=\sum_{C_i\epsilon S}\int_{C_i}\frac{-f}{f+Z}n_1 m ds,$$

$$F_3[X]=\sum_{C_i\epsilon S}\int_{C_i}\frac{-1}{f+Z}(xn_2-yn_1)m ds, \tag{4.11}$$

$$F_4[X]=\sum_{C_i\epsilon S}\int_{C_i}[(\frac{f(Z-Z_0)}{f+Z}-\frac{yY_0}{f+Z}+\frac{y^2}{f})n_1+(\frac{xY_0}{f+Z}-\frac{xy}{f})n_2]m ds,$$

$$F_5[X]=\sum_{C_i\epsilon S}\int_{C_i}[(\frac{yX_0}{f+Z}-\frac{xy}{f})n_1+(\frac{f(Z-Z_0)}{f+Z}-\frac{xX_0}{f+Z}+\frac{x^2}{f})n_2]m ds,$$

$$F_6[X]=\sum_{C_i\epsilon S}\int_{C_i}[(\frac{fX_0}{f+Z}-x)n_1+(\frac{fY_0}{f+Z}-y)n_2]m ds.$$

Since the depth information of the observed object in the first frame is assumed to be given, Z in the above equations can be computed from the image domain.

5. NUMERICAL EXAMPLES

Based on (3.3), there can be several algorithms to recover the motion parameters from the images before and after motion. The first algorithm determines the six motion parameters at the same time at each iteration. However, it is expected that the computation becomes more stable and robust if the number of unknowns is smaller. Noting that translations and rotations are very different types of motion, the "translational scheme" can first be applied, which always sets the rotation to zero when applying (3.3). After computing the estimate of the translation, the image is moved accordingly. Next, the "rotational scheme" is applied, which always sets the translation to zero. After computing the estimate of the rotation, the image is moved accordingly. This process is repeated, applying the translational scheme and the rotational scheme alternately. Furthermore, if the motion of the object is restricted to be on a plane, e.g. cars moving on flat ground, there are only three parameters: two to describe a translation on the plane, and one to describe a rotation around the axis perpendicular to the plane. Again, algorithms to recover the motion parameters may solve for the three parameters simultaneously or use the scheme of alternate translations and rotations.

See [6] for further detail on the experiments. Fig. 1 is an example of using the alternate scheme. The object is moving from the lower left corner to the upper right corner. The true motion parameters are

$$(a, b, c, \Omega_1, \Omega_2, \Omega_3)=(10,10,5,-0.605,0.605,-0.605).$$

The iterations are as follows:

0 (9.100,8.974,2.896,0,0,0)
1 (10.154,9.858,4.814,-0.538,0.776,-0.710)
2 (10.110,9.913,4.903,-0.580,0.679,-0.557)
. . .
9 (10.004,9.995,5.002,-0.601,0.607,-0.601)

6. DISCUSSION

As seen in the above example and in [6], the scheme proposed here can be applied to large motions. In general, convergence is fast for small motions; but the iterations will not converge if the motion is extremely large (especially for rotations by large angles) or if the object contains some facet whose image shrinks sharply when the object moves. However, the alternate application of the translational and rotational schemes can greatly extend the applicable range of the scheme. The assumption of horizontally constrained motion also extends the applicable range. The scheme is quite stable. This may be ascribed to the fact that the initial position is assumed given a priori, and the computation is done only for the motion parameters, while schemes like that of [24] attempt to determine the object position and motion simultaneously.

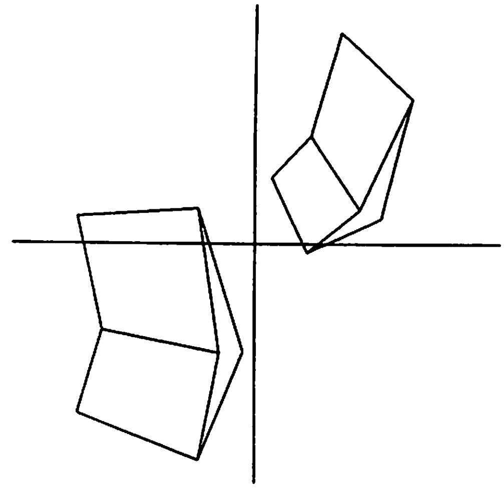

Figure 1. An example of recovering 3D motion.

Note that only one object is considered in the foregoing description of the scheme. In fact, all our equations can be applied to deal with multiple objects, provided these objects are moved "rigidly" together. For example, a camera moving around a static scene is a case of this type.

However, there are still some restriction in applying the method proposed here.

First, complete depth information for the object in the initial frame must be given. Therefore, some other means, say stereo or range sensing, must be employed for the very first frame. However, after that, only one camera is sufficient for using the method since the depth information can be updated from frame to frame by the recovered motion parameters. That is, our method not only recovers the motion parameters but also recovers the structure of the 3D object in successive frames.

On the other hand, consider the case of model-based vision. In this case, the shape of the object of interest is stored along with other model information in a database. Therefore, an image of the object in any position can be generated, and can be used as a reference image. Then, the method proposed here is quite suitable for computing the "motion" which yields a reference image for the observed image. Moreover, much of the computation associated with the reference image can be precomputed and stored in the same database if storage allows. Thus, computation will be faster.

Another limitation is that the same features must be identified in successive frames. Fortunately, the fact that an object normally moves smoothly provides a very good clue for detecting the birth and death of feature primitives, i.e., points, lines, and regions, in the image sequence.

Acknowledgment. The authors thank Profs. Azriel Rosenfeld, Larry S. Davis, and John Aloimonos of the University of Maryland for helpful comments and discussions.

REFERENCES

1. G. Adiv, "Inherent ambiguities in recovering 3-D motion and structure from a noisy flow field", *Proc. DARPA Image Understanding Workshop*, Miami Beach, FL, 1985, 399-412.
2. J. Aloimonos and I. Rigoutsos, "Determining the 3D motion of a rigid surface patch without correspondence under perspective projection", *Proc. AAAI-86*, Philadelphia, PA, Aug. 1986, 681-688.
3. J. Aloimonos and A. Basu, "Shape and 3D motion from contour without point to point correspondences: general principles", *Proc. IEEE Conf. Computer Vision Pattern Recog.*, Miami Beach, FL, June 1986, 518-527.
4. S. Amari, "Invariant structures of signal and feature spaces in pattern recognition problems", *RAAG Memoirs 4* (1968), 553-566.
5. S. Amari, "Feature spaces which admit and detect invariant signal transformations", *Proc. 4th Intl. Joint Conf. Pattern Recog.*, Tokyo, 1978, 452-456.
6. T. C. Chou and K. Kanatani, "Recovering 3D rigid motions without correspondence", Technical Report, Center for Automation Research, University of Maryland, 1987.
7. D. Cyganski and J. A. Orr, "Applications of tensor theory to object recognition and orientation determination", *IEEE Trans. Pattern Anal. Machine Intell. PAMI-7* (Nov. 1985), 662-673.
8. E. Ito and J. Aloimonos, "Determining 3D transformation parameters from images", *Proc. IEEE Conf. Robotics and Automation*, 1987 (to appear).
9. K. Kanatani, "Detecting the motion of a planar surface by line and surface integrals", *CVGIP 29* (1984), 13-22.
10. K. Kanatani, "Dectection of surface orientation and motion from texture by a stereological technique", *Artificial Intelligence 23* (1984), 213-237.
11. K. Kanatani, "Tracing planar surface motion from a projection without knowning the correspondence", *CVGIP 29* (1984), 1-12.
12. K. Kanatani, "Structure from motion without correspondence: general principle", *Proc. DARPA Image Understanding Workshop*, Miami Beach, FL, Dec. 1985, 107-116.
13. K. Kanatani, "Structure from motion without correspondence: general principle", *Proc. Intl. Joint Conf. Artif. Intell.*, Los Angeles, CA, Aug. 1985, 886-888.
14. K. Kanatani and T. C. Chou, "Shape from texture: general principle", *Proc. IEEE Conf. Computer Vision Pattern Recog.*, Miami Beach, FL, June 1986, 578-583.
15. K. Kanatani and T. C. Chou, "Tracing finite motions without correspondence", *Proc. Intl. Workshop on Industrial Applications of Machine Vision and Machine Intelligence*, Tokyo, Feb. 1987.
16. K. Kanatani, "Structure and motion from optical flow under perspective projection", *CVGIP* (to appear).
17. K. Kanatani, "Structure and motion from optical flow under orthographic projection", *CVGIP* (to appear).
18. H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections", *Nature 293* (1981), 133-135.
19. H. C. Longuet-Higgins, "The visual ambiguity of a moving plane", *Proc. Royal Soc. B-223* (1984), 165-175.
20. H. Nagel, "Representation of moving rigid objects based on visual observations", *Computer 14-8* (1981), 29-39.
21. J. W. Roach and J. K. Aggarwal, "Determining the movement of objects from a sequence of images", *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-2* (Nov. 1980), 554-562.
22. A. Rosenfeld and A. C. Kak, *Digital Picture Processing*, Academic Press, New York, 1982.
23. M. Subbarao and A. M. Waxman, "Closed form solution to the image flow equations for planar surfaces in motion", *CVGIP 36* (1986), 208-228.
24. R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of 3D motion parameters of rigid objects with curved surfaces", *IEEE Trans. Pattern Anal. Machine Intell. PAMI-6* (Jan. 1984), 13-27.
25. S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, MA, 1979.
26. A. M. Waxman and K. Wohn, "Contour evolution, neighborhood deformation, and global image flow: planar surfaces in motion", *Int. J. Robotics Res. 4-3* (1985), 95-108.
27. A. M. Waxman and S. Ullman, "Surface structure and three-dimensional motion from image flow kinematics", *Int. J. Robotics Res. 4-3* (1985), 72-94.