# Recovery of 3-D Motion from Time-Varying Image Flows

Kwang Yun Wohn and Soon Ki Jung

## Abstract

In this paper we deal with the problem of recovering 3-D motion and structure from a time-varying 2-D velocity vector field. A great deal has been done on this topic, most of which has concentrated on finding necessary and sufficient conditions for there to be a unique 3-D solution corresponding to a given 2-D motion. While previous work provides useful theoretical insight, in most situations the known algorithms have turned out to be too sensitive to be of much practical use. It appears that any robust algorithm must improve the 3-D solution over time. As a step toward such algorithm, we present a method for recovering 3-D motion and structure from a given time-varying 2-D velocity vector field. The surface of the object in the scene is assumed to be locally planar. It is also assumed that 3-D velocity vectors are piecewise constant over three consecutive frames (or two snapshots of flow field). Our formulation relates 3-D motion and object geometry with the optical flow vector as well as its spatial and temporal derivatives. The linearization parameters, or equivalently, the first-order flow approximation (in space and time) is sufficient to recover rigid body motion and local surface structure from the local instantaneous flow field. We also demonstrate, through a sensitivity analysis carried out for synthetic and natural motions in space, that 3-D motion can be recovered reliably.

## I. General Description of the Method

When an object moves relative to a viewer, the projected image of the object also moves in the image plane. This 2-D image motion reveals the relative 3-D motion of the object as well as the structure of the visible part of that object. By analyzing this evolving image sequence, one hopes to extract the instantaneous 3-D motion (described by six degrees of freedom) and surface geometry of the object. The path from time-varying imagery to its corresponding 3-D description may be divided into two relatively independent steps, although a one-step approach may be possible [15]:

1. computation of 2-D image motion from the image sequence, and
2. computation of 3-D motion and structure of objects from 2-D motion.

This paper deals with the latter issue: computation of 3-D object motion and its structure which undergoes rigid body motion relative to a monocular camera. Throughout this paper, we assume that the 2-D motion is already available in an appropriate representational form, which we will discuss later.

It is well known that the human visual system can solve the problem of recovering 3-D structure from motion information [10, 14, 26]. A number of computational studies have examined possible algorithms for performing this task, assuming 2-D motion measurements are already available. The relations between 2-D motion and 3-D scene are formulated in terms of non-linear equations. This non-linearity prevents us from solving them in a trivial way, which makes the problem mathematically interesting. In some restricted domains (e.g., known angular motion, known object structure) the formulation and solution procedure become much simpler [3, 8, 20].

Since the scheme used to interpret 2-D motion information depends on the kind of 2-D motion representation utilized, the very first consideration is the choice of representation for 2-D image motion. One may use the motion of distinct, well-isolated feature points. Classic work includes Tsai and Huang [22] which proved that seven points all of which lie on a single rigid-body and two perspective views are sufficient in determining their relative position in 3-D space uniquely, and Ullman [24] which showed that three distinct orthographic views of four non-coplanar points yield unique

Manuscript received August 1, 1995; accepted April 8, 1996.

The authors are with Department of Comupter Science, Korea Advanced Institute of Science and Technology.

solution. Variations and extensions are abundant in the motion literature. See [1] for a survey.

The other approach uses the continuous flow field within a small region [16, 17]. Waxman and Ullman showed that the partial derivatives up to the second order of flows revealed the 3-D motion [27]. Later Waxman et. al. devised a closed-form solution for the same formulation [29].

Although 3-D motion is, in principle, recoverable from either representation, it turns out that neither scheme produces the reliable 3-D solution when they are applied to natural images in which various noise effect is not negligible. Those methods perform successfully when the points are well separated (or the size of neighborhood is relatively large), but as can be expected, they encounter severe difficulties when the points get close to each other (or the size of neighborhood gets small) since the image velocities at nearby points are usually nearly identical. While either scheme has its own merits and drawbacks the latter approach (i.e. the optical flow field) leads to stable solutions provided that the partial derivatives of the flow field up to the second-order are available [27]. Waxman and Wohn [28] developed the method of extracting the partial derivatives of the flow field directly from evolving contours over time. The partial derivatives of the flow were called "deformation parameters" since they described the local deformation of a small neighborhood in the image. It has been demonstrated that the combined algorithms of 2-D flow computation and 3-D motion computation are quite stable with respect to input noise and variations in surface structure. Due to the non-linearity associated with the 3-D motion computation, no rigorous analysis on the behavior of this algorithm has been conducted so far, but it appears that the second-order derivatives determine the accuracy of the 3-D solution. However it is very questionable whether these second-order derivatives can be obtained reliably enough so that the 3-D parameters obtained by the algorithm are meaningful, especially when the field of view decreases to under $20^{\circ}$. The approach developed here is based on the first order derivatives of the velocity field. Our experiments confirm that the first-order derivatives can be recovered with greater accuracy than the second-order ones. In our recent experiments conducted on various natural images, we found that the error of first-order derivatives can be reduced to within 5% while the second-order derivatives are obtained with an accuracy of only 30%. However, the knowledge of first-order derivatives at one point in time does not provide enough constraints for solving for 3-D motion and structure. We obtain additional constraints by introducing the partial derivatives of the velocity field with respect to time. The new representational scheme for 2-D motion then consists of a velocity vector, its first-order spatial derivative and its first-order temporal derivative. There are a total of eight parameters, and we shall call them *linearization parameters* of the velocity field, since they are the parameters of the first-order estimate (or linearization) of the velocity vector field. The linearization parameters are obtained from two snapshots of image flow field, or three image frames. However, in practice, one may want to utilize more frames to improve the numerical stability and accuracy.

The idea of utilizing larger number of frames has been proposed by several researchers. Earlier examples include: for a restricted class of rigid-body motion [11], for semi-rigid motion under orthographic projection [25], and for determining the focus of expansion (or constraction) [4]. Recently, more elaborated framework for the recovery of motion during an extended time interval has been proposed. Correlation between successive frames is expressed in terms of the smoothness of 3-D motion, which provides additional motion constraints. In [30] the smoothness of motion was interpreted as the polynomial curve trajectory with constant angular velocity. In [7] both translation and rotation were modeled as the truncated Taylor series. Baker and Bolles [2] seeks for constraints from dense set of spatiotemporal images. Iu and Wohn [12] investigated the 3-D motion of a single feature points.

In our algorithm, the surface of the object in the scene is assumed to be locally planar. It is also assumed that 3-D motion parameters are locally constant over three consecutive frames (or 2 snapshots of the velocity field). Our formulation relates the 3-D motion and object geometry to the 2-D velocity vector and its spatial and temporal derivatives. The linearization parameters, or equivalently, the first-order approximation (in space and time) of the velocity field, is sufficient to recover rigid body motion and local surface structure. We also demonstrate, through a sensitivity analysis carried out for synthetic and natural motions in space, that 3-D inference can be made reliably.

In Section 2, we begin our discussion of the 3-D motion recovery process by establishing a relation which relates the 2-D parameters (linearization parameters of the velocity field) to the parameters of the 3-D object motion and structure. The relationship is formulated in terms of eight non-linear algebraic equations. This formulation requires that the surface of the object be approximated locally by a plane and that 3-D motion parameters do not change over a short period of time. From this point of view there are families of degenerate cases for which the temporal derivatives of velocity field do not provide sufficient information. To resolve these cases the time derivatives of the spatial derivatives may be used. In some other cases, multiple solutions may result due to the non-linearity. The complete solution tree will be presented. We conduct a robustness analysis of the algorithm in Section 4. Of course violation of the initial assumptions on object motion and structure usually result in error in the 3-D solution. The effect of the temporal change of the 3-D motion parameters will be evaluated in Section 4.2. We

demonstrate that such effects should be minor in a range of practical applications. The experimental results on both synthetic data and real time-varying images are presented in Section 5.

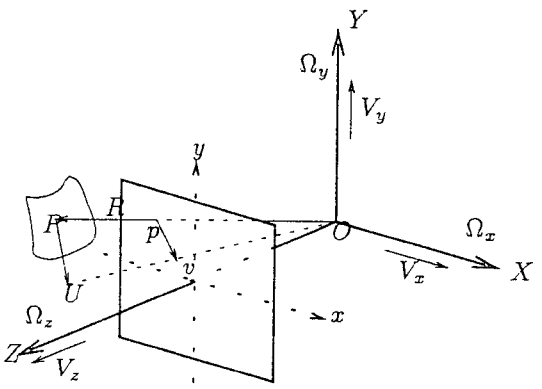



**Fig. 1.** 3-D coordinate system and 2-D image coordinates with motion relative to an object.

## II. Spatio-Temporal Deformation of the Image

In this section we shall give the relations between the 3-D parameters we wish to recover and the measurable parameters of the 2-D motion. The camera model we use is the perspective projection and the coordinate system is the same as the one introduced in [17] (Figure 1). We shall consider only the region near the center of the image plane because we can always choose a proper image coordinate system so that the point being discussed is at the center of the image plane.

### *1. The 3-D and 2-D Parameters*

We represent relative motion in the terms of the viewer's motion. The viewer moves with the translational velocity $\mathbf{V} = [V_X, V_Y, V_Z]^T$, and the rotational velocity $\boldsymbol{\Omega} = [\Omega_X, \Omega_Y, \Omega_Z]^T$. Since the absolute translational velocity is irrecoverable in principle, we are only concerned with recovering the normalized translational velocity

$$(V_X', V_Y', V_Z') \equiv \frac{1}{Z}(V_X, V_Y, V_Z), \tag{1}$$

where $Z$ is the absolute distance from the viewer to the object. When speaking of 3-D structure, we are referring to the surface normal at the point of interest. We represent this as

$$\mathbf{n} \equiv (p, q, 1)/\sqrt{(1+p^2+q^2)} \tag{2}$$

and will work with $(p, q)$.

Let us assume that the image flow $\mathbf{v}(x, y)$ is given. If an image sequence of more than three images is available (or equivalently, if more than two snapshots of the 2-D velocity vector fields are given), we can describe the velocity field as a function of time as well. So we assume we are given a function $\mathbf{v}(x, y, t)$. Taking the partial derivatives of $\mathbf{v}(x, y, t)$, we get six terms

$$\begin{cases} \begin{bmatrix} v_{x,x} & v_{x,y} \\ v_{y,x} & v_{y,y} \end{bmatrix} \equiv \dfrac{\partial \mathbf{v}}{\partial \mathbf{x}} \equiv \begin{bmatrix} \dfrac{\partial v_x}{\partial x} & \dfrac{\partial v_x}{\partial y} \\ \dfrac{\partial v_y}{\partial x} & \dfrac{\partial v_y}{\partial y} \end{bmatrix}, \\ \begin{bmatrix} v_{x,t} \\ v_{y,t} \end{bmatrix} \equiv \dfrac{\partial \mathbf{v}}{\partial \mathbf{x}} \equiv \begin{bmatrix} \dfrac{\partial v_x}{\partial t} \\ \dfrac{\partial v_y}{\partial t} \end{bmatrix}. \end{cases} \tag{3}$$

For convenience we have written $\frac{\partial v_\alpha}{\partial \beta}$ as $v_{\alpha,\beta}$, where the first subscript indicates a component of $\mathbf{v}$ and the second subscript indicates the variable with respect to which a differentiation is performed.

**Caution:** Do not confuse this notation with the notation used for a second-order derivatives, in which no comma appears. We have eight quantities $\{v_x, v_y, v_{x,x}, v_{x,y}, v_{y,x}, v_{y,y}, v_{x,t}, v_{y,t}\}$ — the linearization parameters of the velocity field. They are measurable from (at least) two snapshots of the 2-D velocity fields. We shall see that these eight parameters provide enough constraints to determine $(\mathbf{V}', \boldsymbol{\Omega})$ and $(p, q)$ in the generic case. However, in some special case these eight parameters fail to be independent, and do not provide enough information to permit one to solve for the 3-D parameters. In these cases we may use four additional measurements

$$\begin{bmatrix} v_{x,xt} & v_{x,yt} \\ v_{y,xt} & v_{y,yt} \end{bmatrix} \equiv \frac{\partial^2 \mathbf{v}}{\partial t \partial \mathbf{x}} \equiv \begin{bmatrix} \dfrac{\partial^2 v_x}{\partial t \partial x} & \dfrac{\partial^2 v_x}{\partial t \partial y} \\ \dfrac{\partial^2 v_y}{\partial t \partial x} & \dfrac{\partial^2 v_y}{\partial t \partial y} \end{bmatrix}. \tag{4}$$

Again, the subscript before comma "," indicates a component of $\mathbf{v}$ and the subscripts after the comma indicate the variables with respect to which a differentiation is performed. These quantities can also be obtained from a sequence of more than two snapshots of the velocity field. However, as we have pointed out, these higher-order derivatives recovered may contain larger error than the first-order ones. So we shall not use them unless when it is necessary.

### *2. Spatio-Temporal Deformation Equations*

In order to relate the above parameters to the parameters of the 3-D motion and structure, we shall make two assumptions about the surface of the object and the type of motion:

1. The surface of the object is smooth, and for the object in a fixed position, the surface can be approximated locally by

$$Z(t) = Z_0(t) + p(t)X(t) + q(t)Y(t). \quad (5)$$

2. The motion of the object is smooth. $\mathbf{V}$ and $\boldsymbol{\Omega}$ can be regarded as constants during the short observation period. Since we are using the viewer's motion approach, this means that in a short time period, the viewer moves with constant speed and rotates with constant angular velocity.

These assumptions reflect the spatial smoothness of object surfaces and the temporal smoothness of the motion of physical objects. Any object in the real world is likely to satisfy these two assumptions except for some singular situations, which are beyond of the scope of this paper. Violation of these two assumptions will cause errors. We shall discuss this issue later.

Suppose a point $P$ in space is located at $\mathbf{X}(t) = (X(t), Y(t), Z(t))$, and the coordinate system (viewer) moves with a relative translational velocity $\mathbf{V}$ and rotational velocity $\boldsymbol{\Omega}$. Then

$$\dot{\mathbf{X}}(t) = -[\mathbf{V}(t) + \boldsymbol{\Omega}(t) \times \mathbf{X}(t)], \quad (6)$$

and the corresponding image point has the 2-D velocity vector

$$\begin{cases} v_x(x,y,t) = \left\{x\frac{V_Z(t)}{Z(t)} - \frac{V_X(t)}{Z(t)}\right\} + \{xy\Omega_X(t) - (1+x^2)\Omega_Y(t) + y\Omega_Z(t)\} \\ v_y(x,y,t) = \left\{y\frac{V_Z(t)}{Z(t)} - \frac{V_Y(t)}{Z(t)}\right\} + \{(1+y^2)\Omega_X(t) - xy\Omega_Y(t) - x\Omega_Z(t)\}. \end{cases} \quad (7)$$

These equations define an instantaneous 2-D velocity vector field and assign a unique 2-D vector $\mathbf{v}$ to each image point $(x, y)$ at time $t$. At the center of the image the velocity vector is simply

$$\begin{cases} v_x = -V_X' - \Omega_Y & (8.a) \\ v_y = -V_Y' + \Omega_X. & (8.b) \end{cases} \quad (8)$$

Equations (7) constitute two independent relations among seven unknowns. The various existing techniques for recovering the 3-D parameters differ in the way that they add additional constraints. We shall employ spatial coherence of the velocity field. For this purpose we will have to use our first assumption.

Using Equation (5), differentiating Equations (7) and evaluating them at the origin, we get

$$\begin{cases} v_{x,x} = pV_X' + V_Z' \\ v_{x,y} = qV_X' + \Omega_Z \\ v_{y,x} = pV_Y' - \Omega_Z \\ v_{y,y} = qV_Y' + V_Z'. \end{cases} \quad (9)$$

The left sides of the equations are measurable quantities which represent the relative motion in an infinitesimal neighborhood (one may prefer to say the "eometrical deformation"of the "low field". The above process adds four relations while replacing the unknown $Z$ with two parameters $p$ and $q$. So we now have six independent relations among eight unknowns.

Unlike [17] and [27], in which the additional constraints are obtained by introducing high-order derivatives of the velocity field, we obtain additional constraints by analyzing the way the velocity field changes over time. From Equations (7) one can observe that the velocity changes in time only when

1. The 3-D motion parameters are changing in time, and/or
2. The object distance $Z$ is changing in time.

We have assumed that the 3-D motion parameters do not change during a short time period. To calculate the change in object distance, let us consider how a planar patch moves in 3-D space. From Equation (5) we have

$$dZ_0 - dZ + pdX + Xdp + qdY + Ydq = 0.$$

Substituting Equation (6) into the above equation,

$$\begin{aligned} &\frac{dZ_0}{dt} + V_Z - pV_X - qV_Y - pZ_0\Omega_Y + qZ_0\Omega_X \\ &\quad + \left\{\frac{dp}{dt} - \Omega_Y - q\Omega_Z - p^2\Omega_Y + pq\Omega_X\right\}X \\ &\quad + \left\{\frac{dq}{dt} + \Omega_X + p\Omega_Z + q^2\Omega_X - pq\Omega_Y\right\}Y = 0. \end{aligned}$$

Thus

$$\frac{1}{Z_0}\frac{dZ_0}{dt} = -V_Z' + p(V_X' + \Omega_Y) + q(V_Y' - \Omega_X) \quad (10)$$

$$\begin{cases} \frac{dp}{dt} = \Omega_Y + q\Omega_Z + p^2\Omega_Y - pq\Omega_X \\ \frac{dq}{dt} = -\Omega_X - p\Omega_Z - q^2\Omega_X + pq\Omega_Y. \end{cases} \quad (11)$$

Differentiating Equations (7) with respect to time, evaluating the results at the origin and utilizing Equation (10), we get

$$\begin{cases} v_{x,t} = V_X'\delta \\ v_{y,t} = V_Y'\delta, \end{cases} \quad (12)$$

where

$$\delta \equiv -V_Z' - pv_x - qv_y \quad (13)$$

describes the rate of depth change in time.

Now we have eight equations (8), (9) and (12), with eight unknowns ( $\mathbf{V}$, $\boldsymbol{\Omega}$ and $p$, $q$).

The quantities $v_a, v_{a,\beta}, v_{a,t}$ on the left are measurable from two snapshots of the velocity field. These are the main equations we use in solving the motion problem. As mentioned before, in certain degenerate cases we need to use additional equations. These additional equations are obtained by measuring the temporal change in the spatial derivatives. Differentiating Equations (9) with respect to time, and utilizing Equations (10) and (11), we have

$$\begin{cases} v_{x,xt} = -V_Z'\delta + V_X'm_x \\ v_{x,yt} = V_X'm_y \\ v_{y,xt} = V_Y'm_x \\ v_{y,yt} = -V_Z'\delta + V_Y'm_y, \end{cases} \quad (14)$$

where

$$\begin{cases} m_x = & p\gamma + q\Omega_Z + \Omega_Y \\ m_y = & q\gamma - p\Omega_Z - \Omega_X \\ \gamma = & V_Z' - pV_X' - qV_Y'. \end{cases} \quad (15)$$

The four quantities at the left sides of the equations are also measurable from two snapshots of the image flow field.

## III. Recovery of 3-D Motion

Before we begin to discuss the algorithm for computing 3-D motion, let us briefly mention the problem of estimating 2-D motion in the form of the linearization parameters. In principle, the linearization parameters may be obtained first by recovering the image flow and secondly by taking the partial derivatives of the flow field. But since the differentiation process will amplify noise, we are unlikely to recover these partial derivatives accurately.

In our experiment conducted on the natural time-varying image, contours (interpreted as the continuous edge segments extended over certain length) are used as a primary source of information. However, it is well known that both components of the image flow vector cannot be recovered locally along contours alone [19]. As a contour evolves through an image sequence, one can, in principle, measure the *normal flow* around the contour only, while the motion along the contour is invisible.

Waxman and Wohn developed an algorithm, named *velocity functional method*, that recovers the image flow from evolving contours [28]. The algorithm considers the second-order flow approximation as the starting point, and then computes the best-fitting second-order flow from the local measure of normal flow. As predicted from Equation (7), for planar surfaces ideal data (in other words, perfect measure of normal flow) yields the exact result. For curved surfaces the second-order approximation is not exact and it yields truncation error even for ideal data. This error depends on the size of the field of view (defined by a contour grouping) as well as the geometry (mainly surface curvature) of the corresponding patch.

For our purpose, although the second-order terms will not be used at the later stage of 3-D motion computation, they are included here in order to "absorb" noise, and thereby to obtain less biased spatial derivatives. Temporal derivatives may be obtained by subtracting the spatial parameters over two consecutive image frames. Alternatively, as a better approach, the velocity can be approximated as the truncated Taylor series in the spatio-temporal coordinates;

$$\begin{cases} v_x(x,y,t) = \sum_{i=0}^{2}\sum_{j=0}^{2}\sum_{k=0}^{1} v_x^{(i,j,k)} \frac{x^i}{i!}\frac{y^j}{j!}\frac{t^k}{k!} \\ v_y(x,y,t) = \sum_{i=0}^{2}\sum_{j=0}^{2}\sum_{k=0}^{1} v_y^{(i,j,k)} \frac{x^i}{i!}\frac{y^j}{j!}\frac{t^k}{k!}. \end{cases} \quad (16)$$

The linearization parameters are obtained from the coefficients of polynomials, in a similar manner to the velocity functional method.

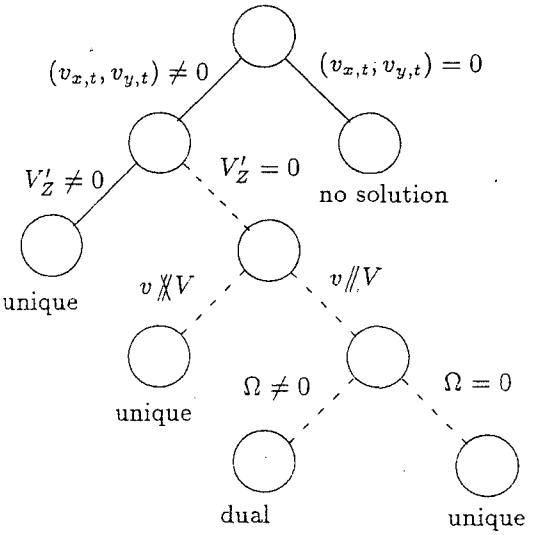



**Fig. 2.** Soulution tree for 3-D structure and motion algorithm.

Once we obtain the linearization parameters, we proceed to recover the 3-D motion and structure of the surface patch. The eight equations to be solved — Equations (8), Equations (9) and Equations (12) — involve eight unknowns. We can prove that in the generic case when $(v_{x,t}, v_{y,t}) \neq 0$ and $V_Z' \neq 0$, we can solve for all the 3-D parameters uniquely from these eight equations. In some other cases these eight equations do not provide enough constraints to determine the 3-D motion and structure. We must use the additional constraints of Equations (14), and in these cases dual solutions may exist. The "worst" case is when all the time derivatives $v_{\alpha,t}$ and $v_{\alpha,\beta t}$ are zero, *i.e.* the velocity vectors in a neighborhood of the center are constant in time: our method fails to recover the 3-D parameters. However, it is possible to resolve the 3-D parameters by knowing the same set of linearization parameters at another point where not all the time derivatives are zero. If $v_{\alpha,t} = v_{\alpha,\beta t} = 0$ for all points in the image, the 3-D motion and structure must be trivial. The detail derivation can be found at Appendix A in [32]. Here we give only the solution tree (Figure 2). Paths indicated by solid lines represent solution procedures requiring only first-order derivatives of the velocity filed (Equations (8), (9) and (12)), while those consisting of dashed lines require second-order derivatives (Equation (14)).

## IV. Robustness of the Algorithm

Our algorithm is "exact" in the sense that if the exact 2-D parameters are given and the two assumptions in Section 2 are satisfied, then the algorithm will produce the exact 3-D solution: There is no search involved. However, in practice

we should never expect all these conditions to be fully satisfied. In this section we are going to discuss what will happen if the two assumptions are violated. The sensitivity of the algorithm with respect to noise in the input 2-D parameters will be demonstrated in the next section.

*1. Error Caused by Non-Planar Surface*

Our analysis is valid for planar surfaces. What about non-planar surfaces? In this section, we argue that non-planarity is not much the determining factor as long as the surface is reasonably smooth, and the computation is performed within a small neighborhood.

Suppose the surface of the object is not planar but still smooth. Then we expand it in a Taylor series:

$$Z = Z_0 + pX + qY + \sum_{n=2}^{\infty}\left[\sum_{i+j=n} a_{ij}X^iY^j\right]. \tag{17}$$

Using Equation (17) instead of Equation (5) and repeating the analysis of Section 2, we obtain exactly the same set of equations as before. This is not surprising since we used only the first order derivatives of the velocity vector field. In theory, then, so long as the surface is smooth, no error will be caused by assuming the surface is planar. In practice, however, since one must consider a small but finite neighborhood in the image plane in order to evaluate the derivatives, non-planarity causes an error if the higher order terms are large. Thus, we require the high order terms to be reasonably small in a small region. Non-planarity will also cause errors in the velocity field during the 2-D motion recovery procedure [28]. However, experiments in the next section show that our algorithm is quite stable with respect to the changes in the surface structure.

*2. Error Caused by Non-Constant Motion*

Since in the preceding sections we assumed that the 3-D motion parameters were constant with respect to time, we shall get additional terms in Equations (12) and (14) involving time derivatives if the real motion is not constant. These terms will cause errors in the final result. In this section we are going to estimate the errors in the 3-D parameters caused by non-constant motion; we will do this only in the generic case, in which only Equations (12) were used.

Suppose acceleration exists:

$$\dot{\mathbf{V}} = \begin{bmatrix} \dot{V}_X \\ \dot{V}_Y \\ \dot{V}_Z \end{bmatrix}, \qquad \dot{\Omega} = \begin{bmatrix} \dot{\Omega}_X \\ \dot{\Omega}_Y \\ \dot{\Omega}_Z \end{bmatrix}.$$

Then Equations (12) become

$$\begin{cases} v_{x,t} = V_X'\delta - dv_{x,t} \\ v_{y,t} = V_Y'\delta - dv_{y,t}, \end{cases} \tag{18}$$

where

$$dv_{x,t} = \frac{\dot{V}_X}{Z} + \dot{\Omega}_Y$$
$$dv_{y,t} = \frac{\dot{V}_Y}{Z} - \dot{\Omega}_X$$

are the additional terms caused by acceleration. Let $\mathbf{v}_t \equiv [v_{x,t}, v_{y,t}]^T$, $d\mathbf{v}_t \equiv [dv_{x,t}, dv_{y,t}]^T$, and define a vector $\mathbf{w} \equiv \mathbf{v}_t + d\mathbf{v}_t$. We can write Equations (18) as

$$\begin{cases} w_x = V_X'\delta \\ w_y = V_Y'\delta. \end{cases} \tag{19}$$

We can see that, if we could measure $w_x$ and $w_y$, then we would have no problem to solve for the 3-D parameters. Since we can only measure $\mathbf{v}_t$, which is only part of $\mathbf{w}$, if we use $\mathbf{v}_t$ as an estimate of $\mathbf{w}$ then the 3-D parameters we obtain will be inaccurate. The size of errors in the 3-D parameter estimates depends on how big the additional terms $dv_{x,t}$ and $dv_{y,t}$ are.

In Appendix B of [32] we give the detailed derivation. The errors on the 3-D parameters turn out to be proportional to $\frac{|d\mathbf{v}_t^{\perp}|}{|\mathbf{v}_t|}$, the "relative change" of $\mathbf{v}_t$ perpendicular to $\mathbf{v}_t$: Acceleration which causes changes in the magnitude of image motion does not affect the solution, whereas acceleration which changes the direction of image motion contaminates the 3-D solution. In the next section we give the experimental results for this type of error.

## V. Experiments

In this section, we verify the result of the sensitivity analysis in Section 4 by conducting a few experiments with synthetic data, and synthetic and real image sequences. Again, the expeiments confirm the robustness of our approach.

*1. Experiment on the Sensitivity of the Algorithm*

Besides the error sources common to any early vision processing, such as errors due to digitization, camera distortion, noise, etc., there are other sources of error related to the particular 2-D motion recovery method being used. The method we used in our experiments is the contour-based method with iterative improvement [33]. In this particular method and any other method that utilizes the evolving edge contours, several factors must be considered: 1) Imperfect contour extraction and false matching, 2) Imperfect normal velocity estimation and 3) Inaccurate motion model. All the factors mentioned above affect the accuracy of the estimate of the velocity field (or the linearization parameters) and therefore affect the 3-D solution. In the first experiment we

test the sensitivity of the algorithm with respect to the noise in the input 2-D parameters by using synthetically generated 2-D motion as the input to the algorithm. The whole experiment procedure is as follows.

1. First, a set of 3-D parameters $P_3=\{p,q,\ \mathbf{V}',\ \Omega\}$, is given.
2. Then a set of 2-D parameters $P_2=\{v_x, v_y, v_{x,x}, v_{x,y}, v_{y,x}, v_{y,y}, v_{x,t}, v_{y,t}\}$ is generated from $P_3$ according to Equations (8), (9) and (12).
3. Define $P_2'=\{u_x, u_y, u_{x,x}, u_{x,y}, u_{y,x}, u_{y,y}, u_{x,t}, u_{y,t}\}$, where

$$\begin{cases} u_\alpha = v_\alpha \\ u_{\alpha,\beta} = v_\alpha + v_{\alpha,\beta}, \end{cases} \qquad \alpha=x,y;\ \beta=x,y,t.$$

Each element $u$ of $P_2'$ is perturbed by rand-om noise to obtain $\tilde{u}$.

$$\tilde{u}=u(1+N_r\nu), \tag{20}$$

where $\nu$ is a random variable uniformly distributed over the interval $[-1,1]$, and $N_r$ is the noise ratio as in Table 1.

**Table 1.** Statistics for 6 simulations.

| $N_r$ | $\frac{\lvert dV\rvert}{\lvert V\rvert}$ | $\frac{\lvert d\Omega\rvert}{\lvert\Omega\rvert}$ | $\frac{\lvert d\,n\rvert}{\lvert n\rvert}$ | $\frac{\lvert dZ_0\rvert}{\lvert Z_0\rvert}$ |
|---|---|---|---|---|
| 0.000000 % | 0.000000 % | 0.000000 % | 0.000000 % | 0.000000 % |
| 1.000000 % | 0.307560 % | 0.311286 % | 0.322219 % | 0.455680 % |
| 2.000000 % | 0.508793 % | 0.550297 % | 0.837837 % | 1.216925 % |
| 3.000000 % | 0.604547 % | 0.716452 % | 1.560563 % | 2.304316 % |
| 4.000000 % | 0.596820 % | 0.809087 % | 2.512627 % | 3.752289 % |
| 5.000000 % | 0.491979 % | 0.827452 % | 3.727105 % | 5.612659 % |

4. From the perturbed $u$'s, a set of noisy 2-D parameters $\tilde{P}_2=\{\tilde{v}_x, \tilde{v}_y, \tilde{v}_{x,x}, \tilde{v}_{x,y}, \tilde{v}_{y,x}, \tilde{v}_{y,y}, \tilde{v}_{x,t}, \tilde{v}_{y,t}\}$ is generated as follows;

$$\begin{cases} \tilde{v}_\alpha = \tilde{u}_\alpha \\ \tilde{v}_{\alpha,\beta} = \tilde{u}_{\alpha,\beta} - \tilde{u}_\alpha, \end{cases} \qquad \alpha=x,y;\ \beta=x,y,t.$$

5. Finally, a set of 3-D parameters $\tilde{P}_3=\{\tilde{p},\tilde{q},\tilde{\mathbf{V}}',\tilde{\Omega}\}$ is obtained from our 3-D motion recovery algorithm by using $\tilde{P}_2$ as input data.

We compare the resulting 3-D parameters obtained from the noisy data and the ideal 3-D parameters by measuring relative errors:

$$\begin{cases} e_{\mathrm{V}} = \dfrac{\lvert \mathbf{V}'-\tilde{\mathbf{V}}'\rvert}{\lvert\tilde{\mathbf{V}}'\rvert} \\ e_\Omega = \dfrac{\lvert\Omega'-\tilde{\Omega}'\rvert}{\lvert\tilde{\Omega}'\rvert} \\ e_{\mathrm{n}} = \dfrac{\lvert \mathrm{n}'-\tilde{\mathrm{n}}'\rvert}{\lvert\tilde{\mathrm{n}}'\rvert}, \end{cases}$$

where n is defined in Equation (2), and $\tilde{\mathrm{n}}=[\tilde{p},\tilde{q},1]$. A typical result is shown in Table 1, where the 3-D parameters used are $\mathbf{V}'=(0.5,0.4,0.3)^T$, $\Omega=(-20^\circ, 10^\circ, 30^\circ)^T$, and $(p,q)=(\tan 30^\circ, \tan 45^\circ)$.

One can see that the error in the result is fairly linear to the error in the input 2-D parameters. We have found that perturbations of about 20 % in the velocity field can be tolerated down to fields of view of about $10^\circ$.

*2. Experiment on Non-Constant Motion*

In the second experiment, we test how non-constant 3-D motion causes errors in the result of our algorithm. (See Appendix B in [32] for the theoretical analysis). In these experiments we add certain accelerations randomly to the translational and rotational velocities to see how they affect the measurement $\frac{\lvert dv_t^{\perp}\rvert}{\lvert v_t\rvert}$ and the 3-D parameter estimates. The accelerations are added in accordance to

$$\dot{\omega}=\omega N_r\nu, \qquad \omega \in \{V_X', V_Y', V_Z', \Omega_X, \Omega_Y, \Omega_Z\},$$

where $\nu$ is the random variable as in Equation (20) and $N_r=5\%$. Three examples are tested.

**Example 1:**

$$\mathbf{V}' = \begin{bmatrix} 0.2 \\ -0.2 \\ -0.3 \end{bmatrix};\quad \Omega = \begin{bmatrix} 10^\circ \\ -10^\circ \\ 5^\circ \end{bmatrix};\quad \begin{bmatrix} p \\ q \end{bmatrix} = \begin{bmatrix} \tan 30^\circ \\ \tan 30^\circ \end{bmatrix},$$

with

$$\begin{bmatrix} \dot{V}_X'/V_X' \\ \dot{V}_Y'/V_Y' \\ \dot{V}_Z'/V_Z' \end{bmatrix} = \begin{bmatrix} 4.1\% \\ -3.7\% \\ -3.6\% \end{bmatrix};\quad \begin{bmatrix} \dot{\Omega}_X/\Omega_X \\ \dot{\Omega}_Y/\Omega_Y \\ \dot{\Omega}_Z/\Omega_Z \end{bmatrix} = \begin{bmatrix} 3.7\% \\ 3.6\% \\ 1.8\% \end{bmatrix}.$$

So

$$\begin{bmatrix} v_{x,t} \\ v_{y,t} \end{bmatrix} = \begin{bmatrix} -0.10 \\ 0.10 \end{bmatrix};\quad \begin{bmatrix} dv_{x,t} \\ dv_{y,t} \end{bmatrix} = \begin{bmatrix} 0.0018 \\ -0.0140 \end{bmatrix},$$

and

$$\frac{\lvert dv_t\rvert}{\lvert v_t\rvert}=9.93\%; \qquad \frac{\lvert dv_t^{\perp}\rvert}{\lvert v_t\rvert}=7.8\%.$$

**Example 2:** For $\mathbf{V}'=(0.1,0.2,0.3)^T$, $\Omega=(30^\circ, -20^\circ, 45^\circ)^T$ and $(p,q)=(\tan -45^\circ, \tan 30^\circ)$, the error propagates to the 2-D measures to yield;

$$\frac{\lvert dv_t\rvert}{\lvert v_t\rvert}=18.67\%; \qquad \frac{\lvert dv_t^{\perp}\rvert}{\lvert v_t\rvert}=11.81\%.$$

**Example 3:** For $\mathbf{V}'=(-0.2,-0.2,0.3)^T$, $\Omega=(10^\circ, -10^\circ, 10^\circ)^T$ and $(p,q)=(\tan 30^\circ, \tan 45^\circ)$, the error propagates to the 2-D measures to yield;

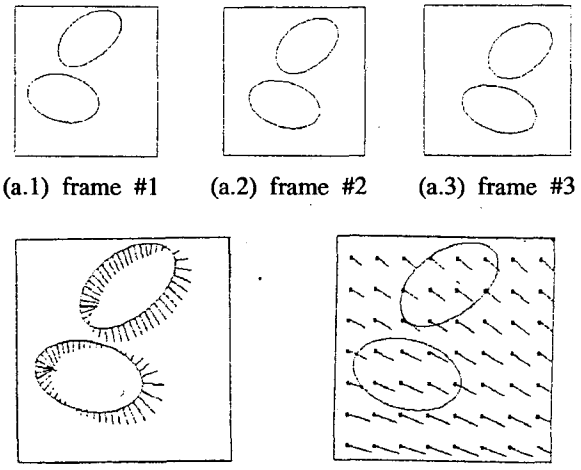
$$\frac{\lvert dv_t\rvert}{\lvert v_t\rvert}=3.85\%; \qquad \frac{\lvert dv_t^{\perp}\rvert}{\lvert v_t\rvert}=3.77\%.$$

The errors caused by non-constant motion are about the same order or magnitude as $\frac{\lvert dv_t^{\perp}\rvert}{\lvert v_t\rvert}$.

*3. Experiment on Synthetic Images*

Next, we test the algorithm when edge contours are extracted perfectly. We still need to compute the linearization

parameters from the edge sequence before we solve for 3-D motion. In this experiment three synthetically generated images were used. The frames are of size 256 by 256 pixels, and the scene is of two ellipses on a plane defined by $Z=10+pX+qY$ with $V'=(0.06, 0.03, -0.04)^T$, $\Omega=(0^\circ, 0^\circ, 0^\circ)^T$ and $(p, q)=(\tan 30^\circ, \tan 45^\circ)$. The images of the ellipses are shown in Figure 3(a). Normal velocities along the contours were measured from the pair of consecutive images (Figure 3(b)). The iterative 2-D motion recovery procedure [33] using the image flow model of Equation (16) was used to obtain the velocity field as of Figure 3(c).



(a.1) frame #1 (a.2) frame #2 (a.3) frame #3

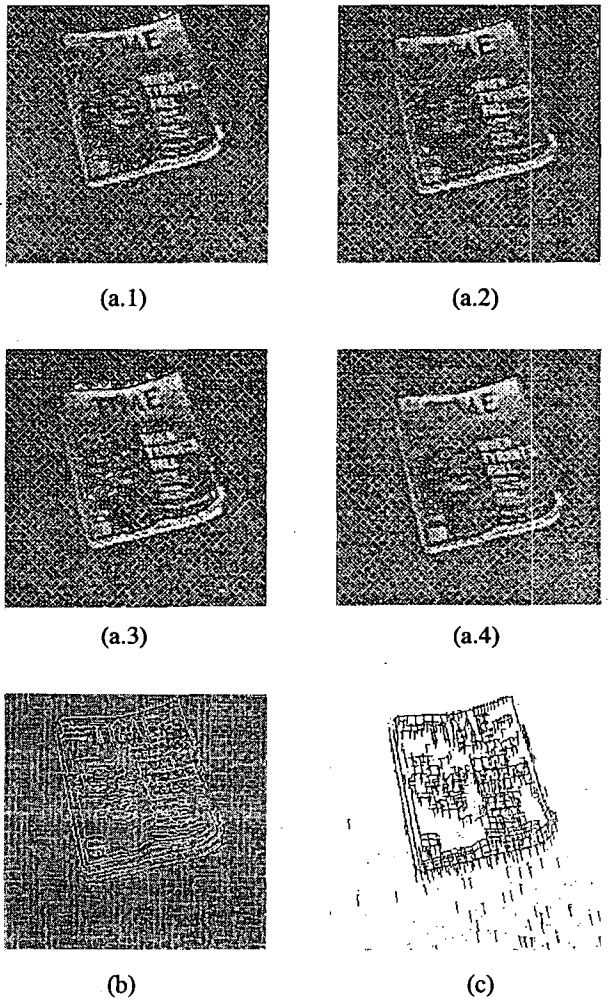
(b) (c)

**Fig. 3.** Recovering optical flow from evolving contours. (a) contours on the image planes as input (3 frames are shown). (b) measured normal flow along the contours (from frame #1 and #2). (c) optical flow field recovered.

The linearization parameters were recovered with 7% of error. The 3-D parameters computed from these velocity fields are: $V'=(0.059, 0.028, -0.038)^T$, $\Omega=(-0.0021^\circ, 0.00064^\circ, -0.00071^\circ)^T$ and $(p, q)=(\tan 31.87^\circ, \tan 46.26^\circ)$, which yields $e_V=2.8\%$ and $e_n=4.7\%$.

### *4. Experiment on Real Images*

Figure 4(a) shows four consecutive images obtained from a natural scene. A CCD camera with known view angle and focal length was attached to a robot arm so that the motion could be controlled. The images were 512 by 512 pixels in size with 8 bits per pixel, but were subsampled to 256 by 256 for computational consideration. The motion parameters and the orientation of the object are $V'=(0.065, 0.031, -0.023)^T$ at the first frame, $\Omega=(0^\circ, 0^\circ, 0^\circ)^T$, and $(p, q)=(-\tan 5.4^\circ, -\tan 24.8^\circ)$. V and $\Omega$ were kept constant, but because of the change in $Z$ the relative translation $V'$ changed. The evolution of zerocrossing contours from the first frame to the second frame is illustrated in Figure 4(b). Figure 4(c) shows the 2-D velocity vector fields recovered (along zerocrossing contours) using the iterative motion recovery procedure [33] with the second-order flow model over the entire filed of view. Unlike for the previous experiments, the temporal derivatives were obtained by subtracting the spatial parameters over two successive frames. The 3-D parameters computed from these are $V'=(0.0057, 0.035, 0.0022)^T$, $\Omega=(-0.75^\circ, 0.53^\circ, -0.087^\circ)^T$, and $(p, q)=(-\tan 2.5^\circ, -\tan 21.7^\circ)$. $e_T$, the relative error in the translational velocity is 12.8%. The error in the rotational velocity is within $1^\circ$, while the surface orientation was obtained within $3^\circ$ of error. A few other experiments under the similar setting have been conducted resulting in the similar outcome.



(a.1) (a.2)

(a.3) (a.4)

(b) (c)

**Fig. 4.** Experiment on real images. (a) Four frames of input image. (b) Superimposed zerocrossing contours. The contours of frame 1 are shown in white, and those of frame 2 are shown in black. (c) Full flow recovered.

### *5. Summary of Experiments*

The above experiments suggest that this method is quite

robust. The error in 3-D motion parameters is proportional to the error in the linearization parameters. In an unstructured environment which generates images whose complexity is comparable with the images we have used in this paper, the linearization parameters could be recovered within 10% of error. On the other hand, second-order derivatives of image flow which other *Eulerian* approaches [21, 27] rely on are totally useless under the presence of noise and digitization effects.

Although many independent factors affect the accuracy of the 3-D solution, we have found, through various experiments, that the temporal derivatives $v_{x,t}$ and $v_{y,t}$ play the major role. The performance drops as the temporal derivatives reduce, and finally when the temporal derivatives become near zero, the method switches to another branch of solution tree (Case 2) looking for additional constraints from the higher-order image formation. In this case, the algorithm performs as well (or bad) as the existing methods [21, 27]. Other noteworthy observations are: 1) The method breaks down when the field of view (or the size of neighborhood in which the linearization parameters are evaluated) is smaller than $6\sim8^\circ$ . This is consistent with the simulation result reported in [28]. The method of [28] utilized the second-order deformation and broke down when the field of view decreased to $12^\circ$ . 2) 3-D motion is recovered quite accurately when there is a large translational motion along Z-axis. This is consistent with the experimental result in [9] for the same reason.

## VI. Concluding Remarks

We have presented an algorithm which recovers 3-D structure and motion from the linearization parameters — the parameters of the first-order estimate of the velocity vector field. In most cases, these parameters can be recovered quite reliably. We carried out a sensitivity analysis for synthetic and natural motions in space, to demonstrate that 3-D motion can also be recovered reliably. 3-D motion was further refined progressively in time by using the ordinary differential equations which describe the evolution of motion and structure.

In a typical unstructured environment 2-D motion (in terms of the linearization parameters) can be obtained within 10 % of error, which in turn produces the 3-D motion parameters within the same range of accuracy. The method behaves well when the viewing angle (or the size of local neighborhood) is larger than $10^\circ$ , but breaks down when the viewing angle gets smaller than $6\sim8^\circ$ . This is due to the fact that the variation of flow field is not visible in a sufficiently small aperture, often masked by other adversary effects such as digitization and the error in edge localization.

## References


[1] J. K. Aggarwal and N. Nandhakumar, "On the computation of motion from sequences of images — A review," *Proc. IEEE*, Vol. 76, pp. 917-935, 1988.

[2] H. H. Baker and R. C. Bolles, "Generalizing Epipolar-plane image analysis on the spatiotemporal surface," *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 2-9, Ann Arbor, June 1988.

[3] D. H. Ballard and O. A. Kimball, "Rigid body motion from depth and optical flow," *Computer Vision, Graphics, and Image Processing*, Vol. 22, No. 1, pp. 95-115, Apr. 1983.

[4] A. Bandyopadhay and J. Aloimonos, "Perception of rigid motion from spatio-temporal derivatives of optical flow," *Computer Science Department TR-157*, Univ. Rochester, Mar. 1985.

[5] T. J. Broida and R. Chellappa, "Estimation of object motion parameters from noisy images," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-8, No. 1, pp. 90-99, Jan. 1986.

[6] T. J. Broida and R. Chellappa, "Kinematics and structure of a rigid object from a sequence of noisy images," *Proc. IEEE Workshop on Motion*, pp. 95-100, 1986.

[7] T. J. Broida and R. Chellappa, "Experiments and uniqueness results on object structure and kinematics from a sequence of monocular images," *IEEE Workshop on Visual Motion*, pp. 21-30, Mar. 1989.

[8] A. R. Bruss and B. K. P. Horn, "Passive navigation," *Computer Vision, Graphics, and Image Processing*, Vol. 21, pp. 3-20, 1983.

[9] J. Q. Fang and T. S. Huang, "Some experiments on estimating the 3-D motion parameters of a rigid body from two consecutive image frames," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, No. 5, pp. 547-554, Sep. 1984.

[10] J. J. Gibson, *The Senses Considered as Perceptual Systems*, Houghton Mifflin, Boston, MA, 1966.

[11] D. D. Hoffman, "Inferring local surface orientation from motion fields," *Journal Optical Soc. Am.*, Vol. 72, pp. 888-892, 1982.

[12] S. L. Iu and K. Wohn, "Estimation of 3-D motion and structure based on a temporally oriented approach with the method of regression," *IEEE Workshop on Visual Motion*, pp. 273-281, Mar. 1989, Irvine, CA.

[13] S. L. Iu and K. Wohn, "Model mismatch and usage of the FLAT MMF for the motion of a single particle," *Grasp Lab Tech Report*, Feb. 1990.

[14] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception and Psychophysics*, Vol. 14, pp. 201-211, 1973.

[15] K. Kanatani, "Detection of surface orientation and

motion from texture by a stereological technique," *Artificial Intelligence*, Vol. 23, pp. 213-237, 1984.

[16] J. J. Koenderink and A.J. van Doorn, "Invariant properties of the motion parallax field due to the movement of rigid bodies relatives to an observer," *Optica Acta*, Vol. 22, pp. 773-791, 1975.

[17] H. C. Longuet-Higgins and K. Prazdny, "The interpretation of a moving retinal image," *Proc. Royal Soc. London B*, Vol. 208, pp. 385-397, 1980.

[18] D. Marr and E. C. Hildreth, "Theory of edge detection," *Proc. Royal Soc. London B*, Vol. 207, pp. 187-217, 1980.

[19] D. Marr and S. Ullman, "Directional sensitivity and its use in early visual processing," *Proc. Royal Soc. London B*, Vol. 211, pp. 151-180, 1981.

[20] J. H. Rieger and D. T. Lawton, "Sensor motion and relative depth from difference fields of optical flows," *Proc. 8th Int. Joint Conf. Artificial Intell.*, Aug. 1983.

[21] M. Subbarao, "Interpretation of image flow: Rigid curved surfaces in motion," *Intl. Journal Computer Vision*, Vol. 2, No. 1, pp.77-96, June 1988.

[22] R. Y. Tsai and T. S. Huang, "Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, No. 1, pp. 13-27, Jan. 1984.

[23] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE Trans. Robotics Automation*, Vol. 3, No. 4, Aug. 1987.

[24] S. Ullman, *The Interpretation of Visual Motion*, MIT Press, Cambridge, 1979.

[25] S. Ullman, "Maximizing rigidity: the incremental recovery of 3-D structure and rubbery motion," *M.I.T. A.I. Memo 721*, June 1983.

[26] H. Wallach and D. N. O'Connell, "The kinetic depth effect," *Journal Exp. Psych.*, Vol. 45, pp. 205-217, 1953.

[27] A. M. Waxman and S. Ullman, "Surface structure and 3-D motion from image flow: kinematic analysis," *Intl. Journal of Robotics Research*, Vol. 4, pp. 72-94, 1985.

[28] A. M. Waxman and K. Wohn, "Contour evolution, neighborhood deformation and global image flow: Planar Surfaces in Motion," *Intl. Journal of Robotics Research*, Vol. 4, pp. 95-108, 1985.

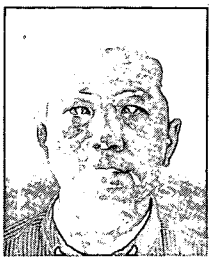
[29] A. M. Waxman, B. K-Parsi, and M. Subbarao, "Closed-form solutions to image flow equations for 3D structure and motion," *Intl. Journal of Computer Vision*, Vol. 1, No. 3, pp. 239-258, 1987.

[30] J. Weng, T.S. Huang, and N. Ahuja, "3-D motion estimation, understanding, and prediction from noisy image sequences," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-9, No. 3, pp. 370-389, May 1987.

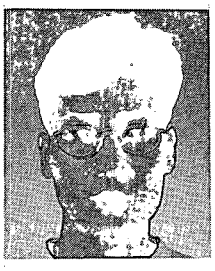
[31] K. Wohn and J. Wu, "3-D motion recovery from time-varying optical flows," *Proc. AAAI-86*, pp. 670-675, Aug. 1986.

[32] K. Wohn and S. K. Jung, "Recovery of 3-D motion from time-varying image flows," *VR-TM-95-22*, CSD KAIST, July 1995.

[33] J. Wu, R. Brockett, and K. Wohn, "A contour-based recovery of image flow: iterative method," *Proc. IEEE Conf. Computer Vision Pattern Recognition*, pp. 124-129, San Diego, June 1989.



**Kwang-yun Wohn** was born in Korea, on February 28, 1952. He received B.S. degree in Applied Physics from Seoul National University, Seoul, Korea, in 1974, and the M.S. degree from the University of Wisconsin, Madison in 1981, and the Ph.D. degree in Computer Science from University of Maryland, College Park in 1984, respectively. From 1974 to 1979, he was a Research Engineer of the Agency for Defence Development (ADD). From 1984 to 1986, he was a lecturer at Division of Applied Science, Harvard University. In 1986 ~ 1990, he joined the Department of Computer and Information Science at University of Pennsylvania as an Assistant Professor. Since 1990, he has been with the Department of Computer Science at Korea Advanced Institute of Science and Technology (KAIST) as an Assistant Professor. His research interests are virtual reality, artificial intelligence, computer vision, image processing, computer art and computational music.



**Soon-Ki Jung** was born in Korea, on September 23, 1967. He received the B.S. degree from Kyungbook National University, Taegu, Korea, in 1990, and the M.S. degree in Computer Science from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1992. He is currently working toward the Ph.D. degree in Computer Science at KAIST. His research interests include computer graphics, computer vision, and virtual reality.