A Simplified Linear Optic Flow-Motion Algorithm

Xinhua Zhuang, Thomas S. Huang, and Narendra Ahuja

Coordinated Science Laboratory, University of Illinois, 1101 W. Springfield Avenue, Urbana, Illinois 61801

AND

Robert M. Haralick

Department of Electrical Engineering, University of Washington, Seattle, Washington 98195

The purpose of the article is to establish the relationship between rigid body motion and the optic flow image and solve motion parameters from optic flow image points. A basic equation relating optic flow image points to rigid body motion which involves only the instantaneous rotation and translation velocities without depths is established. A linear algorithm is developed to determine the mode of motion (whether the instantaneous translation is zero or not), the instantaneous rotation velocity, the direction of the instantaneous translation velocity, and the relative depth map (or surface structure) under the rank assumption. The algorithm represents a simplification to the linear optic flow-motion algorithm proposed in (Zhuang and Haralick, in *Proceedings, IEEE First Conf. on Artificial Intelligence Applications, Denver, CO, 1984*, pp. 366–375). © 1988 Academic Press, Inc.

I. INTRODUCTION

The purpose of the article is to establish the relationship between rigid body motion and its corresponding optic flow perspective projections and solve motion parameters from optic flow image points. The optic flow perspective projection is contained in the optic flow image which for each pixel (X, Y) contains the projected motion (u, v).

In the article a basic equation relating optic flow image points and rigid body motion (ω, k) with ω being the instantaneous rotation velocity and k being the instantaneous translation velocity is established (see Section II). The equation called the *optic flow-motion equation* does not involve in-depth information. From the optic flow-motion equation, a unique instantaneous rotation velocity ω and a nonzero vector k^* which is colinear with the instantaneous translation velocity k could be determined under the *rank assumption* (see Section III). The rank assumption requires that at least eight (or six) optic flow image points be observed when the instantaneous translation velocity k is nonzero (or zero). The rank assumption will be violated when, for instance, the observed optic flow image points all come from the same rigid planar patch. A simple way to determine the mode of motion (i.e., whether the instantaneous translation velocity is zero or not), the direction of translation, and the surface structure (i.e., the relative depth map) when the translation is not zero, is shown in Section IV. A linear algorithm which represents a simplification to the previous one in [1] is given in Section V. The simulation results are also included there. The final section is a summary.

334

The algorithm is stable. Based on it a robust algorithm is being developed in order to reduce the effects caused by wrongly computed optic flow image points and segment multiple rigid body motions.

II. OPTIC FLOW-MOTION EQUATION

Suppose a rigid body is in motion in the half space, $z < 0$. Let $p(t)$ be the position vector of an object point at the time t, $p(t) = (x(t), y(t), z(t))^T$ where T represents the transposition operation. Let $(X(t), Y(t))$ denote the central projective coordinates of $p(t)$ onto the image plane, $z = 1$:

$$X(t) = x(t)/z(t), \qquad Y(t) = y(t)/z(t),$$
$$p(t) = z(t)(X(t), Y(t), 1)^T. \tag{1}$$

Let $(u(t), v(t))$ denote the instantaneous velocity of the moving image point $(X(t), Y(t))$, i.e., $(u(t), v(t)) = (\dot{X}(t), \dot{Y}(t))$, where the overdot represents the time derivative.

We shall call each element $[(X(t), Y(t)), (u(t), v(t))]$ as an optic flow image point. Usually, we may not be able to observe all optic flow image points. Denote a finite number of observed optic flow image points by $q_i = [(X_i(t), Y_i(t)), (u_i(t), v_i(t))]$, $i = 1, \ldots, n$.

The instantaneous representation of the rigid motion is described by

$$\dot{p}(t) = \omega(t) \times p(t) + k(t). \tag{2}$$

where $\omega(t) = (\omega_1(t), \omega_2(t), \omega_3(t))^T$ represents the instantaneous rotational angular velocity of the rigid motion and $k(t) = (k_1(t), k_2(t), k_3(t))^T$ the instantaneous translational velocity.

Differentiating (1), we could express $\dot{p}$ in (2) as

$$\dot{p} = \dot{z}(Z, Y, 1)^T + z(u, v, 0)^T, \tag{3}$$

where for simplicity the time variable "t" has been omitted. Combining (2) and (3), it follows that

$$\dot{z}(X, Y, 1)^T + z(u, v, 0)^T = z\omega \times (X, Y, 1)^T + k, \tag{4}$$

from which we could derive the optic flow-motion equation. As a matter of fact, for any nonzero vector k^* which is colinear with k (i.e., $k^* \times k = 0$) taking its cross product with both sides of (4) yields

$$\dot{z}k^* \times (X, Y, 1)^T + zk^* \times \left[(u, v, 0)^T - \omega \times (X, Y, 1)^T\right] = 0. \tag{5}$$

Since z is assumed nonzero, Eq. (5) indicates that either the vector $k^* \times [(u, v, o)^T - \omega \times (X, Y, 1)^T]$ equals zero or it is parallel to $k^* \times (X, Y, 1)^T$. In the first case, the vector k^* is parallel to $[(u, v, o)^T - \omega \times (X, Y, 1)^T]$. The latter case implies that $k^* \times (X, Y, 1)^T$ is orthogonal to $[(u, v, o)^T - \omega \times (X, Y, 1)^T]$. In any case,

$$\langle k^* \times (X, Y, 1)^T, (u, v, 0)^T - \omega \times (X, Y, 1)^T \rangle = 0, \tag{6}$$

or

$$-\langle k^* \times (X, Y, 1)^T, \omega \times (X, Y, 1)^T\rangle + \langle (u, v, o)^T, k^* \times (X, Y, 1)^T\rangle = 0. \quad (7)$$

Let

$$K^* = \begin{bmatrix} 0 & -k_3^* & k_2^* \\ k_3^* & 0 & -k_1^* \\ -k_2^* & k_1^* & 0 \end{bmatrix}, \quad \Omega = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}. \quad (8)$$

Then

$$k^* x(X, Y, 1)^T = K^*(X, Y, 1)^T,$$
$$\omega x(X, Y, 1)^T = \Omega(X, Y, 1)^T. \quad (9)$$

And Eq. (7) could be written as

$$\langle (X, Y, 1)^T, K^*\Omega(X, Y, 1)\rangle + \langle (u, v, o)^T, K^*(X, Y, 1)^T\rangle = 0, \quad (10)$$

or in a more symmetric form,

$$\langle (X, Y, 1)^T \frac{K^*\Omega + \Omega K^*}{2}(X, Y, 1)^T\rangle + \langle (u, v, o)^T, K^*(X, Y, 1)^T\rangle = 0, \quad (11)$$

which does not involve the depth z and is called the *optic flow-motion equation*.

In the next section we show a necessary and sufficient condition called the *rank assumption* under which a unique rotation ω and a nonzero vector k^*, which is colinear with k, could be determined simply from the optic flow-motion equation.

III. A LINEAR ALGORITHM FOR SOLVING OPTIC FLOW-MOTION EQUATION

Let L be $(K^*\Omega + \Omega K^*)/2$ which is just the symmetric part of the matrix $K^*\Omega$. Then we have n linear homogeneous equations with unknowns L and K^*:

$$\langle (X_i, Y_i, 1)^T, L(X_i, Y_i, 1)^T\rangle + \langle (u_i, v_i, o)^T, K^*(X_i, Y_i, 1)^T\rangle = 0, \quad i = 1, \ldots, n. \quad (12)$$

It is easy to see that Eq. (12) contains nine unknowns (six for L and three for K^*). It is also clear that being colinear with k, k^* equals either αk with α an arbitrary real number when $k \neq 0$ or an arbitrary 3×1 vector when $k = 0$. In other words, being the solution of (12), the pair (L, K^*) will have one degree of freedom when $k \neq 0$ or three degrees of freedom when $k = 0$. Thus, (L, K^*) which is defined by ω and k^* compromises the general solution to the system of n linear homogeneous equations, i.e., Eq. (12), if and only if the $n \times 9$ coefficient matrix of Eq. (12) has a

rank eight when $k \neq 0$ or a rank six when $x = 0$. Let

$$L = \begin{bmatrix} l_1 & l_4 & l_5 \\ l_4 & l_2 & l_6 \\ l_5 & l_6 & l_3 \end{bmatrix},$$

$$h = \left(l_1, l_2, l_3, 2l_4, 2l_5, 2l_6, k_1^*, k_2^*, k_3^* \right)^{\mathrm{T}},$$

$$a_i = \left(X_i^2, Y_i^2, 1, X_i Y_i, X_i, Y_i, -v_i, u_i, v_i X_i - u_i Y_i \right)^{\mathrm{T}}.$$

(13)

then Eq. (12) could be written in a more readable form,

$$a_i^{\mathrm{T}} h = 0, \qquad i = 1, \ldots, n, \tag{14}$$

which are n linear homogeneous equations whose least-squares solution up to a scale factor is given by

$$\min_{\|h\|=1} h^{\mathrm{T}} \left(\sum_{i=1}^{n} a_i a_i^{\mathrm{T}} \right) h. \tag{15}$$

That is, the solution to Eq. (14) coincides with the eigenvector of smallest eigenvalue of the 9×9 symmetric and nonnegative matrix $W = \sum_{i=1}^{n} a_i a_i^{\mathrm{T}}$ whose rank is the same as the $n \times 9$ coefficient matrix, i.e.,

$$\begin{bmatrix} a_1^{\mathrm{T}} \\ \vdots \\ a_n^{\mathrm{T}} \end{bmatrix}_{n \times 9},$$

of Eq. (14). Thus, h, which is defined by ω and k^* through L and K^*, coincides with the smallest eigenvalue-vector of W if and only if the following rank assumption holds:

$$\begin{array}{ll} \mathrm{Rank}(W) = 8 & \text{when } k \neq 0, \\ \mathrm{Rank}(W) = 6 & \text{when } k = 0. \end{array} \tag{16}$$

Since each optic flow image point contributes only one row to the coefficient matrix of Eq. (14), to assure (16) we need at least eight optic flow image points ($n \geq 8$) when $k \neq 0$ or six points ($n \geq 6$) when $k = 0$. More optic flow points are preferable to smooth out the effect of any noise. From now on we assume (16).

Once a nonzero smallest eigenvalue-vector h of W is determined, L and $K^*(\neq 0)$ can then be determined. The rotation ω could be uniquely determined through the equation $L = (K^*\Omega + \Omega K^*)/2$. The latter is equivalent to

$$\omega_1 k_1^* = \frac{l_1 - l_2 - l_3}{2}, \tag{17}$$

$$\omega_2 k_2^* = \frac{l_2 - l_3 - l_1}{2}, \tag{18}$$

$$\omega_2 k_3^* = \frac{l_3 - l_1 - l_2}{2}, \tag{19}$$

$$\omega_2 k_1^* + \omega_1 k_2^* = 2l_4, \tag{20}$$

$$\omega_1 k_3^* + \omega_3 k_1^* = 2l_5, \tag{21}$$

$$\omega_2 k_3^* + \omega_3 k_2^* = 2l_6. \tag{22}$$

When $k_1^* \neq 0$, Eqs. (17), (20), (21) could be used to determine ω. When $k_2^* \neq 0$, Eqs. (18), (20), (22) could be used to determine ω. And when $k_3^* \neq 0$, Eqs. (19), (21), (22) could be used to determine ω. For stability, we recommend the following scheme to uniquely determine ω:

Step 1. If $0 \neq |k_1^*| \geq |k_2^*|, |k_3^*|$, then use Eqs. (17), (20), (21).

Step 2. Otherwise, if $|k_2^*| \geq |k_3^*|$, then use Eqs. (18), (20), (22).

Step 3. Otherwise, use Eqs. (19), (21), (22).

Therefore, under the rank assumption (16) the optic flow-motion equation uniquely determines the instantaneous rotation of ω and a nonzero vector k^* which is colinear with k. As mentioned before, k^* equals αk when $k \neq 0$ or could be an arbitrary nonzero vector when $k = 0$. Apparently, ω and k^* are all we can get from the optic flow-motion equation.

In the next section we show how to simply determine the mode of motion (i.e., whether $k = 0$ or not), the direction of k, and the surface structure, i.e., the relative depths when $k \neq 0$.

IV. MODE OF MOTION, DIRECTION OF TRANSLATION, AND SURFACE STRUCTURE

From Eq. (4) we see that

$$\dot{z} = z \begin{bmatrix} \omega_1 & \omega_2 \\ X & Y \end{bmatrix} + k_3. \tag{23}$$

Using Eq. (23) to eliminate $\dot{z}$ in Eq. (4), it follows that

$$z \left\{ \begin{bmatrix} \omega_1 & \omega_2 \\ X & Y \end{bmatrix} (X, Y)^{\mathrm{T}} - \left(\begin{bmatrix} \omega_2 & \omega_3 \\ Y & 1 \end{bmatrix}, \begin{bmatrix} \omega_3 & \omega_1 \\ 1 & X \end{bmatrix} \right)^{\mathrm{T}} + (u, v)^{\mathrm{T}} \right\}$$
$$+ \left\{ k_3 (X, Y)^{\mathrm{T}} - (k_1, k_2)^{\mathrm{T}} \right\} = 0. \tag{24}$$

Let

$$f_i(\omega) = \begin{bmatrix} \omega_1 & \omega_2 \\ X_i & Y_i \end{bmatrix} (X_i, Y_i)^{\mathrm{T}} - \left(\begin{bmatrix} \omega_2 & \omega_3 \\ Y_i & 1 \end{bmatrix}, \begin{bmatrix} \omega_3 & \omega_1 \\ 1 & X_i \end{bmatrix} \right)^{\mathrm{T}},$$
$$g_i(k) = k_3 (X_i, Y_i)^{\mathrm{T}} - (k_1, k_2)^{\mathrm{T}}. \tag{25}$$

Then we obtain n equations of the form (24) as follows:

$$z_i \left\{ f_i(\omega) + (u_i, v_i)^{\mathrm{T}} \right\} + g_i(k) = 0, \qquad i = 1, \ldots, n. \tag{26}$$

As is easily seen, $k = 0$ if and only if for at least two points $g_i(k) = 0$. Thus, since $z_i \neq 0$, $k \neq 0$ if and only if for all except possibly one point $f_i(\omega) + (u_i, v_i)^{\mathrm{T}} \neq 0$. This means that the mode of motion could be uniquely determined. When $k \neq 0$ is confirmed, the direction of translation and the surface structure could be simply determined as follows: From Eq. (26) we obtain

$$\left\{ f_i(\omega) + (u_i, v_i)^{\mathrm{T}} \right\} z_i / \|k\| + g_i(k/\|k\|) = 0, \tag{27}$$

$$\|f_i(\omega) + (u_i, v_i)^{\mathrm{T}}\|^2 z_i / \|k\| + \langle f_i(\omega) + (u_i, v_i)^{\mathrm{T}}, g_i(k/\|k\|) \rangle = 0. \tag{28}$$

Since all z_i's are negative, it is clear from Eq. (28) that $\langle f_i(\omega) + (u_i, v_i)^{\mathrm{T}}, g_i(k/\|k\|)\rangle$ is positive except possibly one point where it is zero. It is also clear that $k/\|k\|$ equals $k^*/\|k^*\|$ or $-k^*/\|k^*\|$ (depending on whether the translation k has the same direction as k^* or $-k^*$) and correspondingly

$$\langle f_i(\omega) + (u_i, v_i)^{\mathrm{T}}, g_i(k/\|k\|)\rangle = \pm\langle f_i(\omega) + (u_i, v_i)^{\mathrm{T}}, g_i(k^*/\|k^*\|)\rangle. \quad (29)$$

Thus, combining Eqs. (28) and (29) it follows that

$$z_i/\|k\| = -\left|\langle f_i(\omega) + (u_i, v_i)^{\mathrm{T}}, g_i(k^*/\|k^*\|)\rangle\right|\Big/\left\|f_i(\omega) + (u_i, v_i)^{\mathrm{T}}\right\|^2. \quad (30)$$

Furthermore, the translation k has the same direction as k^* or $-k^*$ if and only if $\sum_i\langle f_i(\omega) + (u_i, v_i)^{\mathrm{T}}, g_i(k^*)\rangle$ is positive or negative.

As seen from the previous and current sections there exists a linear algorithm to uniquely determine ω, the mode of motion, the direction of translation, and the relative depth map or surface structure $z_i/\|k\|$, $i = 1, \ldots, n$, when $k \neq 0$. In the next section we give that algorithm which is simpler than the one in [1] and some simulation results.

V. ALGORITHM AND SIMULATION RESULTS

Now comes the following algorithm:

Step 1. Compute $\min_{\|h\|=1} h^t W h$.
　　　　　Set $k^* = (h_7, h_8, h_9)^{\mathrm{T}}$.
Step 2. If $(|k_1^*| \geq |k_2^*|, |k_3^*|)$, then set
　　　　　$\omega_1 = (h_1 - h_2 - h_3)/(2k_1^*)$,
　　　　　$\omega_2 = (h_4 - k_2^*\omega_1)/k_1^*$,
　　　　　$\omega_3 = (h_5 - k_3^*\omega_1)/k_1^*$.
Step 3. Otherwise, if $(|k_2^*| \geq |k_3^*|)$, then set
　　　　　$\omega_2 = (h_2 - h_3 - h_1)/(2k_2^*)$,
　　　　　$\omega_3 = (h_6 - k_3^*\omega_2)/k_2^*$,
　　　　　$\omega_1 = (h_4 - k_1^*\omega_2)/k_2^*$.
Step 4. Otherwise, set
　　　　　$\omega_3 = (h_3 - h_1 - h_2)/(2k_3^*)$,
　　　　　$\omega_1 = (h_5 - k_1^*\omega_3)/k_3^*$,
　　　　　$\omega_2 = (h_6 - k_2^*\omega_3)/k_3^*$.
Step 5. If $(\sum_i\|f_i(\omega) + (u_i, v_i)^{\mathrm{T}}\| = 0)$, then set
　　　　　$k = 0$. Stop.
Step 6. Otherwise, set
　　　　　$$z_i/\|k\| = -\left|\langle f_i(\omega) + (u_i, v_i)^{T}, g_i(k^*/\|k^*\|)\rangle\right|\Big/\left\|f_i(\omega) + (u_i, v_i)^{T}\right\|^2.$$
Step 7. The translation k has the same direction as k^* or $-k^*$ if $\sum_i\langle f_i(\omega) + (u_i, v_i)^{\mathrm{T}}, g_i(k^*)\rangle$ is positive or negative.
　　　　　Stop.

Simulation Results. The experiments needed to verify the above algorithm should tell us: (1) what is the minimum number of points in order to compute

ZHUANG ET AL.

TABLE 1

The Noiseless Case

True direction of translation	Estimated direction of translation	True rotation	Estimated rotation	x	y	u	v
0.74	0.74	−1.59	−1.59	$0.38e + 00$	$−0.90e + 00$	$−0.23e + 02$	$−0.57e + 01$
0.58	0.58	1.34	1.34	$−0.77e − 01$	$−0.19e + 00$	$−0.61e + 01$	$−0.28e + 01$
−0.34	−0.34	−4.41	−4.41	$−0.12e + 01$	$0.42e + 00$	$−0.31e + 01$	$−0.54e + 01$
				$−0.27e + 01$	$−0.27e + 01$	$0.23e + 02$	$0.56e + 02$
				$−0.64e + 00$	$−0.50e + 00$	$−0.79e + 01$	$−0.11e + 01$
				$−0.41e + 00$	$0.20e + 01$	$−0.13e + 02$	$−0.38e + 02$
				$−0.54e + 00$	$−0.75e + 00$	$−0.87e + 01$	$0.91e + 00$
				$−0.26e + 01$	$0.51e + 00$	$0.20e + 02$	$−0.38e + 02$
0.74	0.74	−1.59	−1.59	$0.32e + 00$	$−0.45e + 00$	$−0.65e + 01$	$−0.81e + 01$
0.58	0.58	1.34	1.34	$0.30e + 00$	$−0.24e + 00$	$−0.53e + 01$	$−0.30e + 01$
−0.34	−0.34	−4.41	−4.41	$−0.72e + 00$	$−0.69e + 00$	$−0.51e + 01$	$0.28e + 01$
				$−0.67e − 01$	$0.40e + 00$	$−0.35e + 01$	$−0.44e + 01$
				$−0.17e + 01$	$−0.61e + 00$	$0.77e − 01$	$0.25e + 01$
				$−0.70e + 00$	$−0.62e + 00$	$−0.77e + 01$	$0.27e + 00$
				$0.50e + 00$	$0.53e + 00$	$−0.26e + 01$	$−0.56e + 01$
				$−0.10e + 00$	$−0.65e + 01$	$0.12e + 03$	$0.28e + 03$
0.74	0.74	−1.59	−1.59	$−0.19e + 00$	$−0.35e + 00$	$−0.59e + 01$	$−0.13e + 01$
0.58	0.58	1.34	1.34	$−0.10e + 00$	$0.42e + 00$	$−0.15e + 01$	$−0.25e + 01$
−0.34	−0.34	−4.41	−4.41	$0.32e + 00$	$−0.34e + 00$	$−0.11e + 02$	$−0.57e + 01$
				$0.14e + 01$	$−0.19e + 01$	$−0.43e + 02$	$−0.44e − 01$
				$−0.12e + 00$	$0.75e − 01$	$−0.29e + 01$	$0.25e + 01$
				$−0.23e + 00$	$−0.56e + 00$	$−0.79e + 01$	$0.27e + 00$
				$0.69 + 00$	$0.64e + 00$	$−0.98e + 01$	$−0.56e + 02$
				$0.58e + 00$	$−0.44e + 00$	$−0.20e + 02$	$0.28e + 02$
0.74	0.74	−1.59	−1.59	$−0.85e + 00$	$0.18e + 01$	$−0.38e + 02$	$−0.11e + 03$
0.58	0.58	1.34	1.34	$−0.26e + 00$	$0.36e + 00$	$−0.66e + 01$	$−0.74e + 01$
−0.34	−0.34	−4.41	−4.41	$0.35e + 00$	$0.51e + 00$	$−0.19e + 01$	$−0.45e + 01$
				$−0.82e + 00$	$−0.59e − 01$	$−0.42e + 01$	$−0.24e + 01$
				$−0.20e + 01$	$0.26e + 01$	$0.83e + 01$	$−0.37e + 02$
				$0.31e + 00$	$0.40e + 00$	$−0.31e + 01$	$−0.49e + 01$
				$0.11e + 00$	$0.83e + 00$	$−0.14e + 02$	$−0.19e + 02$
				$0.77e + 00$	$−0.20e + 00$	$−0.14e + 02$	$−0.97e + 01$
0.74	0.74	−1.59	−1.59	$0.12e + 01$	$0.46e + 00$	$−0.12e + 02$	$−0.14e + 02$
0.58	0.58	1.34	1.34	$−0.27e + 00$	$0.45e + 00$	$−0.24e + 01$	$−0.34e + 01$
−0.34	−0.34	−4.41	−4.41	$−0.51e + 00$	$0.95e + 00$	$−0.22e + 01$	$−0.70e + 01$
				$−0.68e + 00$	$−0.84e + 00$	$−0.62e + 01$	$0.33e + 01$
				$0.13e + 01$	$−0.34e + 00$	$−0.45e + 02$	$−0.23e + 02$
				$−0.89e − 01$	$0.53e + 00$	$−0.13e + 01$	$−0.29e + 01$
				$0.10e + 01$	$0.16e + 01$	$−0.23e + 02$	$−0.33e + 02$
				$0.89e + 00$	$−0.12e + 00$	$−0.12e + 02$	$−0.96e + 01$

TABLE 2

The Noise Case: $|u - \bar{u}| + |v - \bar{v}| \leq 0.2$, Where (u, v) Is True Optic Flow Velocity
and $(\bar{u}, \bar{v})$ Is Erroneous Optic Flow Velocity

True direction of translation	Estimated direction of translation	True rotation	Estimated rotation	x	y	u	v
− 0.69	− 0.68	0.42	0.43	$0.43e - 01$	$-0.30e - 01$	$0.95e + 01$	$0.74e + 01$
− 0.53	− 0.54	− 1.15	− 1.16	$0.13e + 01$	$0.55e + 00$	$0.24e + 02$	$0.16e + 02$
0.49	0.49	0.22	0.17	$-0.13e + 01$	$0.82e + 00$	$-0.19e + 01$	$0.20e + 02$
				$-0.19e + 00$	$-0.13e + 01$	$0.14e + 02$	$-0.40e + 01$
				$0.24e + 00$	$-0.37e + 00$	$0.51e + 01$	$0.23e + 01$
				$-0.32e + 00$	$0.30e + 00$	$0.43e + 01$	$0.68e + 01$
				$-0.96e - 01$	$-0.38e - 01$	$0.35e + 01$	$0.33e + 01$
				$-0.25e + 00$	$-0.49e + 00$	$0.18e + 02$	$0.93e + 01$
− 0.69	− 0.69	0.42	0.42	$0.64e + 00$	$0.45e + 00$	$0.91e + 01$	$0.76e + 01$
− 0.53	− 0.54	− 1.15	− 1.15	$0.10e + 00$	$-0.44e + 00$	$0.12e + 02$	$0.52e + 01$
0.49	0.48	0.22	0.22	$-0.33e + 00$	$0.31e + 00$	$0.29e + 01$	$0.50e + 01$
				$0.45e + 00$	$0.44e + 00$	$0.70e + 01$	$0.64e + 01$
				$0.54e + 00$	$0.87e + 00$	$0.19e + 02$	$0.20e + 02$
				$-0.21e + 02$	$0.20e + 02$	$-0.21e + 05$	$0.86e + 04$
				$-0.17e + 02$	$-0.16e + 02$	$-0.39e + 04$	$-0.36e + 04$
				$0.68e + 00$	$-0.65e + 00$	$0.11e + 02$	$0.26e + 01$
− 0.69	− 0.67	0.42	0.43	$-0.11e + 01$	$-0.11e + 00$	$0.96e - 01$	$0.66e + 01$
− 0.53	− 0.55	− 1.15	− 1.16	$0.10e + 00$	$-0.12e + 01$	$0.16e + 02$	$-0.21e + 01$
0.49	0.49	0.22	0.16	$0.10e + 00$	$0.36e - 01$	$-0.69e + 01$	$0.56e + 01$
				$-0.30e + 00$	$0.54e + 00$	$0.81e + 01$	$0.13e + 02$
				$-0.25e + 00$	$0.21e + 01$	$0.18e + 02$	$0.51e + 02$
				$0.25e + 01$	$-0.20e + 01$	$0.68e + 02$	$-0.12e + 02$
				$0.31e + 01$	$-0.28e + 01$	$0.34e + 03$	$-0.12e + 03$
				$0.16e + 00$	$0.17e + 01$	$0.44e + 02$	$0.80e + 02$
− 0.69	− 0.68	0.42	0.41	$0.11e + 01$	$0.11e + 00$	$0.16e + 02$	$0.87e + 01$
− 0.53	− 0.54	− 1.15	− 1.14	$0.12e + 00$	$-0.47e + 00$	$0.43e + 01$	$0.18e + 01$
0.49	0.49	0.22	0.26	$-0.50e + 00$	$0.30e + 00$	$0.28e + 01$	$0.61e + 01$
				$-0.58e + 00$	$-0.58e + 00$	$0.20e + 01$	$0.12e + 01$
				$-0.21e + 02$	$-0.80e + 01$	$-0.51e + 04$	$-0.18e + 04$
				$-0.82e + 00$	$-0.21e + 01$	$0.93e + 01$	$-0.24e + 02$
				$0.50e + 00$	$0.55e + 00$	$0.63e + 01$	$0.61e + 01$
				$-0.21e + 00$	$0.54e + 00$	$0.42e + 01$	$0.70e + 01$
− 0.69	− 0.67	0.42	0.43	$-0.049e + 00$	$0.41e + 00$	$0.17e + 01$	$0.47e + 01$
− 0.53	− 0.53	− 1.15	− 1.16	$-0.38e + 00$	$-0.20e + 01$	$0.14e + 02$	$-0.16e + 02$
0.49	0.50	0.22	0.17	$-0.16e + 00$	$0.80e + 00$	$0.71e + 01$	$0.12e + 02$
				$0.24e + 01$	$0.21e + 01$	$0.60e + 02$	$0.50e + 02$
				$0.59e + 00$	$0.43e + 00$	$0.72e + 01$	$0.61e + 01$
				$0.86e + 02$	$-0.11e + 03$	$0.61e + 05$	$-0.76e + 05$
				$-0.31e + 02$	$-0.31e + 02$	$-0.91e + 04$	$-0.93e + 04$
				$0.94e + 00$	$0.15e + 01$	$0.29e + 02$	$0.33e + 02$

motion and surface structure from accurate or noisy optic flow measurements in practice; (2) what is the likelihood we come across a set of optic flow image points that violate the rank assumption (i.e., (16)); and (3) what is the accuracy of the estimated motion parameters from noisy optic flow measurements.

One way of testing the validity of the rank assumption is to generate many sets of data randomly and to compute the eigenvalues of W. If W has fewer than eight (six) nonzero eigenvalues when $k \neq 0$ $(= 0)$, then the rank assumption is violated. This method, however, suffers from the defect that, since most eigenvalue computing routines return inexact values even with accurate given data, we need a threshold around zero below which we would like to say the eigenvalues are zero or nearly zero. This obviously is a problem, since the rank assumption is not violated until more than one (three) eigenvalues are strictly zero when $k \neq 0$ $(= 0)$.

The approach used here has therefore been different. We have actually implemented the algorithm and tested it with many sets of randomly chosen data. Implicitly, this tells us the likelihood of the occurrence of sets of optic flow image points that violate the rank assumption, since any time we encounter such a set of points, the algorithm will break down. It also answers questions 1 and 3.

Several observations from our experiments are as follows:

In spite of testing the algorithm with an extremely large number of sets of randomly generated points, we have not met a set of points that violate the rank assumption.

For the case where there is no noise in the optic flow measurements, the algorithm is extremely accurate. Even when there is noise in the optic flow measurements, the algorithm works well except that the mode of motion cannot be determined correctly. That is, the case of pure rotation (i.e., $k = 0$) cannot be recognized. If the case of pure rotation is wrongly recognized as the general case (i.e., $k \neq 0$), a relative depth map and a direction of translation will be given by the algorithm. Since relative depth and translation direction cannot be determined in the case of pure rotation, what the algorithm gives will not cause any conflict to any given information. Thus, acceptance of the interpretation extracted by the algorithm could never be a disaster as long as the noise is not large.

For the sake of space, only a few experimental results are tabulated here (see Tables 1, 2).

VI. SUMMARY

There are a number of interesting problems related to optic flow. How to compute optic flow reliably and efficiently is of basic importance (see [2]–[10]). As is known, the problem is ill-posed in general. Without further cues, reliable optic flow image points can be obtained only at prominent image feature points like corners. How to segment the image into areas containing differently moving objects is another difficult and crucial problem (see [11]–[12]) which usually involves occlusion analysis. Finally, how to determine the motion and the surface structure from optic flow, is perhaps, a little bit easier in comparison with the previous two problems (see [1, 13–23]). This article provides a simplification to the earlier proposed linear optic flow-motion algorithm in [1].

Similar to the two-view motion algorithms (see [24–26]), optic flow-motion algorithms still need a matching process in order to acquire the optic flow image

points. However, the matching process in the optic flow case could proceed in each small neighborhood. This often makes things easier even though the matching remains unsolved in general.

Some other approaches are possible. For instance, in the planar patch case the method of using image intensity derivatives instead of optic flow image points to recover motion were reported in [27, 28]. We are currently working on a line-based algorithm to recover motion and surface structure.

REFERENCES

1. X. Zhuang and R. M. Haralick, Rigid body motion and the optic flow image, in *Proceedings, IEEE First Conf. on Artificial Intelligence Applications, Denver, CO, 1984*, pp. 366–375.
2. B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artif. Intell.* **17**, 1981, 185–203.
3. K. Woh, H. Xie, and L. S. Davis, Smoothing optical flow fields, in *Proceedings, Image Understanding Workshop, Washington, DC, 1983*, pp. 61–63.
4. R. Paquin and E. Dubois, A spatio-temporal gradient method for estimating the displacement field in time-varying imagery, *Comput. Vision Graphics Image Process.* **21**, 1983, 205–221.
5. A. Mitiche, Computation of optical flow and rigid motion, in *Proceedings, Workshop on Computer Vision: Representation and Control, Annapolis, MD, 1984*, pp. 63–71.
6. E. C. Hildreth and S. Ullman, *The Measurement of Visual Motion*, MIT Press, Cambridge, MA, 1983.
7. S. T. Barnard and W. B. Thompson, Disparity analysis in images, *IEEE Trans. Pattern Anal. Mach. Intell.* **2**, 1980, 333–340.
8. J. K. Kearney and W. B. Thompson, Gradient-based estimation of optical flow with global optimization, in *Proceedings, IEEE First Conf. on Artificial Intelligence Applications, Denver, CO, 1984*, pp. 376–380.
9. H. H. Nagel, Displacement vectors derived from second-order intensity variation in image sequences, *Comput. Vision Graphics Image Process.* **21**, 1983, 85–117.
10. D. W. Murray and B. F. Buxton, Reconstruction of the optic flow field from edge motion: An examination of two different approaches, in *Proceedings, IEEE First Conf. on Artificial Intelligence Applications, Denver, CO, 1984*, pp. 382–388.
11. C. L. Fennema and W. B. Thompson, Velocity determination in scenes containing several moving objects, *Comput. Graphics Image Process.* **9**, 1979, 301–315.
12. G. Adiv, Determining three dimensional motion and structure from optical flow generated by several moving objects, *IEEE Trans. Pattern Anal. Mach. Intell.* **7**, No. 4, 1985, 384–401.
13. B. Neumann, Motion analysis of image sequences for object grouping and reconstruction, in *Proceedings, Int. Conf. on Pattern Recognition, Miami, FL, Dec. 1980*, pp. 1262–1265.
14. H. C. Longuet-Higgins and K. Prazdny, The interpretation of moving retinal images, *Proc. R. Soc. London Ser. B* **208**, 1980, 385–397.
15. K. Prazdny, On the information in optical flows, *Comput. Vision Graphics Image Process.* **22**, 1983, 239–259.
16. A. M. Waxman, An image flow paradigm, in *Proceedings, Workshop on Computer Vision, Annapolis, MD, 1984*.
17. A. M. Waxman and S. Ullman, *Surface Structure and 3D Motion from Image Flow*, CAR-TR-24, Center for Automation Research, University of Maryland, Oct. 1983.
18. C. Jerian and R. Jain, Determining motion parameters for schemes with translation and rotation, *IEEE Trans. Pattern Anal. Mach. Intell.* **6**, No. 4, pp. 523–529, 1984.
19. J. H. Rieger and D. H. Lawton, Determining the instantaneous axis of translation from optical flow generated by arbitrary sensor motion, in *Proceedings, Workshop on Motion: Representation and Perception, Toronto, Canada, April 1983*, pp. 33–41.
20. J. Aloimonas and C. M. Brown, Direct processing of curvilinear sensor motion from a sequence of perspective images, in *Proceedings, Workshop on Computer Vision, Annapolis, MD, 1984*.

21. R. M. Haralick and X. Zhuang, A note on "Rigid body motion from depth and optical flow," *Comput. Vision Graphics Image Process.* **34**, 1986, 372–387.

22. W. W. Martin and J. K. Aggarwal, Dynamic scenes and object descriptions, in *Proceedings, Int. Conf. Acoustics, Speech, and Signal Processing, Paris, May 3–5, 1982*, pp. 859–862.

23. 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**, No. 6, 1980, 554–562.

24. Longuet-Higgins, A computer algorithm for reconstructing a scene from two projections, *Nature* **293**, Sept. 1981, 133–135.

25. R. Y. Tsai and T. S. Huang, Uniqueness and estimation of 3-D motion parameters of rigid objects with curved surfaces, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAMI-6**, No. 1, 1984, 13–26.

26. X. Zhuang, T. S. Huang, and R. M. Haralick, Two-view motion analysis: A unified algorithm, *J. Opt. Soc. Amer. Ser. A* **3**, No. 9, 1986, 1492–1500.

27. R. Y. Tsai and T. S. Huang, Estimating three-dimensional motion parameters of a rigid planar patch, *IEEE Trans. Acoust. Speech Signal Process.* **ASSP-29**, No. 6, 1981, 1147–1152.

28. S. Negahdaripour and B. K. P. Horn, Direct passive navigation: Analytical solution for planes, in *Proceedings, Workshop on Image Understanding, Miami Beach, FL, Dec. 1985*, pp. 381–387.