Determining 3-D Motion of Planar Objects from Image Brightness Patterns

S. Negahdaripour and B.K.P. Horn

Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

Abstract: The brightness patterns in two successive image frames are used to recover the motion of a planar object without computing the optical flow as an intermediate step. Based on a least-squares formulation, a set of nine nonlinear equations are derived. A simple iterative scheme for solving these equations is presented. Using a selected example, it is shown that in general, the scheme may converge to cither of two possible solutions depending on the initial condition. Only in the special case where the translational motion vector is perpendicular to the surface does our algorithm converge to a unique solution.

1. Introduction

The problem of determining rigid object motion and surface structure from a sequence of image frames has been the topic of many recent research papers in the area of machine vision. Much of the theoretical work has been restricted to using the optical flow, the apparent velocity of brightness patterns in the image Three types of approaches, discrete, differential, and least-squares, have been commonly pursued.

In the discrete approach, information about a finite number of points is used to reconstruct the motion [3,7,11-13]. To do this, one has to identify and match feature points in a sequence of images. The minimum number of points required depends on the number of image frames. In the differential approach, one uses the optical flow and its first and second derivatives at a single point [8,15]. In the least-squares approach, the optical flow is used at every image point [1,2,16].

In general, to compute the optical flow, one exploits a constraint equation between the optical flow and the image brightness gradients. Locally, the brightness variations in time varying images only provide one constraint on the two components of the optical flow. Therefore, an additional constraint will be required to compute the local flow field. For instance, one may assume that the flow field varies smoothly [5,6], or that it is locally quadratic [15].

In this paper, we restrict ourselves to planar surfaces where only three parameters are needed to specify the surface structure. We determine the motion and surface parameters directly from the image brightness gradients, without having to compute the optical flow as an intermediate step.

2. Problem Formulation

Horn and Schunk [6] have derived a constraint equation between the optical flow (u, v), the apparent velocity of brightness patterns in the image, and the spatial-temporal gradients of the brightness patterns (E_x, E_y, E_t), when the incident illumination is uniform across the surface. This constraint equation is of the form:

$$E_x u + E_y v + E_t = 0. \tag{1}$$

In practice, the brightness gradients are estimated from the gray levels in consecutive image frames using finite difference methods.

Any rigid body motion can be decomposed into translational and rotational components. We can either consider the motion of an object relative to a stationary camera, or equivalently the motion of a camera relative to a stationary object (navigation). In either case, the relative motion between the object and the camera and the object structure are to be determined from sequences of image frames. Let $\mathbf{t} = (U, V, W)^T$ and $\omega = (A, B, C)^T$ denote the vectors of translational and rotational velocity, respectively (^{T} denotes the transpose of a vector), and let the point $\mathbf{r} = (x, y, 1)^T$ in the image plane be the perspective projection of the point $\mathbf{r} = (X, Y, Z(X, Y))^T$ on the rigid object. It can be shown that the optical flow generated in the image plane by the relative motion between the camera and the object is given by[2]:

$$\begin{aligned} u &= A\,xy - B(x^2+1) + C\,y + (-U + xW)/Z, \\ v &= A\,(y^2+1) - B\,xy - C\,x + (-V + yW)/Z. \end{aligned} \tag{2}$$

Substituting equations (2) into (1), and simplifying the results, we obtain the brightness change constraint equation for the case of rigid body motion:

$$c + \mathbf{v}\cdot\omega + \frac{1}{Z}\,\mathbf{s}\cdot\mathbf{t} = 0, \tag{3}$$

where $c = E_t$, and

$$\mathbf{v} = \begin{pmatrix} E_x xy + E_y(y^2+1) \\ -E_x(x^2+1) - E_y xy \\ E_x y - E_y x \end{pmatrix}, \quad \mathbf{s} = \begin{pmatrix} -E_x \\ -E_y \\ E_x x + E_y y \end{pmatrix}.$$

For a planar surface, $(\mathbf{r}\cdot\mathbf{n})\ Z = 1$, where $\mathbf{n} = (p,q,r)^T$ is the normal to the surface. Substituting for Z into equation (3) yields:

$$c + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t}) = 0. \tag{4}$$

This is the brightness change constraint equation for a rigid planar object undergoing 3-D motion. We will exploit it to recover the motion and surface parameters.

3. Least-Squares Formulation

Given perfect data, only a few points are sufficient to determine the nine unknowns (three components of w, t, and n)—or rather, eight, since we can only recover the distance to the plane and the translational velocity up to a scale factor (Equation (4) will remain invariant if n is multiplied by a scale factor and t is divided by the same factor). In practice, this constraint equation will not be satisfied at each image point due to additive sensor noise, quantization of the image brightness levels and finite difference approxaimation used to estimate the brightness gradients. Therefore, a least-squares formulation seems appropriate in developing a robust algorithm.

A suitable choice of surface and motion parameters should minimize some measure of error in equation (4) for every image point. We will formulate the following unconstrained optimization problem:

Find the surface n, and motion parameters *w* and t, that minimize the expression:

$$J = \iint_{\Omega} [c + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t})]^2\, dx\, dy, \tag{5}$$

where the integration is performed over the relevant region Ω of the image plane. Necessary conditions for minimizing equation (5) with respect to $\boldsymbol{\omega}$, $\mathbf{t}$, and $\mathbf{n}$ include:

$$\frac{\partial J}{\partial \boldsymbol{\omega}} = \mathbf{0}, \quad \frac{\partial J}{\partial \mathbf{t}} = \mathbf{0}, \quad \text{and} \quad \frac{\partial J}{\partial \mathbf{n}} = \mathbf{0}. \tag{6}$$

Performing the indicated differentiations in (6) we get:

$$\iint_{\Omega} [c + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t})]\,\mathbf{v}\, dx\, dy = \mathbf{0}, \tag{7a}$$

$$\iint_{\Omega} (\mathbf{r}\cdot\mathbf{n})\,[c + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t})]\,\mathbf{s}\, dx\, dy = \mathbf{0}, \tag{7b}$$

$$\iint_{\Omega} (\mathbf{s}\cdot\mathbf{t})\,[c + \mathbf{v}\cdot\boldsymbol{\omega} + (\mathbf{r}\cdot\mathbf{n})(\mathbf{s}\cdot\mathbf{t})]\,\mathbf{r}\, dx\, dy = \mathbf{0}. \tag{7c}$$

These comprise nine nonlinear simultaneous equations that can be solved for the six motion parameters, $\mathbf{t}$ and $\boldsymbol{\omega}$, and the three surface parameters, $\mathbf{n}$.

4. An Iterative Solution Procedure

We now present an iterative solution procedure for solving the nine simultaneous equations defined previously. First, some observations about equations (7) are in order:

1. Equation (7a) is linear in $\boldsymbol{\omega}$, $\mathbf{t}$, and $\mathbf{n}$.
2. Equation (7b) is linear in $\boldsymbol{\omega}$ and $\mathbf{t}$, but quadratic in $\mathbf{n}$.
3. Equation (7c) is linear in $\boldsymbol{\omega}$ and $\mathbf{n}$, but quadratic in $\mathbf{t}$.

Several iterative schemes for solving equations (7) can be considered. The two we have implemented are as follows:

1. Solve the linear equations in (7a) and (7b) for $\boldsymbol{\omega}$ and $\mathbf{t}$ in terms of $\mathbf{n}$, and the linear equations in (7c) for $\mathbf{n}$ in terms of $\boldsymbol{\omega}$ and $\mathbf{t}$ in an iterative procedure.
2. Solve the linear equations in (7a) and (7b) for $\boldsymbol{\omega}$ and $\mathbf{t}$ in terms of $\mathbf{n}$, and the linear equations in (7a) and (7c) for $\boldsymbol{\omega}$ and $\mathbf{n}$ in terms of $\mathbf{t}$ in an iterative procedure.

The second scheme involves more computation, but converges faster. We will only describe the first scheme here (see [10] for more details on both schemes).

Expanding equations (7), collecting terms, and simplifying the results yield:

$$\begin{pmatrix} \mathbf{M}_1 & \mathbf{M}_2 \\ \mathbf{M}_2^T & \mathbf{M}_4 \end{pmatrix} \begin{pmatrix} \boldsymbol{\omega} \\ \mathbf{t} \end{pmatrix} = -\begin{pmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{pmatrix}, \tag{8}$$

$$\mathbf{N}\mathbf{n} = -\mathbf{g}, \tag{9}$$

where

$$\mathbf{M}_1 = \iint_{\Omega} (\mathbf{v}\mathbf{v}^T)\, dx\, dy \quad \mathbf{M}_2 = \iint_{\Omega} \mathbf{n}^T\mathbf{r}(\mathbf{v}\mathbf{s}^T)\, dx\, dy$$

$$\mathbf{M}_4 = \iint_{\Omega} \mathbf{n}^T(\mathbf{r}\mathbf{r}^T)\mathbf{n}(\mathbf{s}\mathbf{s}^T)\, dx\, dy$$

$$\mathbf{N} = \iint_{\Omega} \mathbf{t}^T(\mathbf{s}\mathbf{s}^T)\mathbf{t}(\mathbf{r}\mathbf{r}^T)\, dx\, dy$$

$$\mathbf{d}_1 = \iint_{\Omega} c\mathbf{v}\, dx\, dy \quad \mathbf{d}_2 = \iint_{\Omega} c(\mathbf{s}\mathbf{r}^T)\mathbf{n}\, dx\, dy$$

$$\mathbf{g} = \iint_{\Omega} \left[c(\mathbf{r}\mathbf{s}^T)\mathbf{t} + \mathbf{t}^T\mathbf{s}(\mathbf{r}\mathbf{v}^T)\boldsymbol{\omega}\right] dx\, dy.$$

Given the surface parameters $\mathbf{n}$, the motion parameters $\boldsymbol{\omega}$ and $\mathbf{t}$ can be determined from equation (8). Similarly, given the motion parameters $\boldsymbol{\omega}$ and $\mathbf{t}$, equation (9) can be solved for the surface parameters $\mathbf{n}$. Based on these observations, we adopt the following iterative scheme:

1. Start with an initial guess for the surface parameters.
2. Solve the matrix equation (8) for motion parameters.
3. Solve the matrix equation (9) for surface parameters.
4. Evaluate the improvement in the solution to either go to (2) for the next iteration or stop if the solution has not improved.

The solution of equation (8) for $\boldsymbol{\omega}$ and $\mathbf{t}$ can be determined analytically, and can be written in many forms. For example, if $\mathbf{M}_1$ is invertible, we have

$$\mathbf{t} = \left(\mathbf{M}_4 - \mathbf{M}_2^T\mathbf{M}_1^{-1}\mathbf{M}_2\right)^{-1}\left(\mathbf{M}_2^T\mathbf{M}_1^{-1}\mathbf{d}_1 - \mathbf{d}_2\right), \tag{10a}$$

$$\boldsymbol{\omega} = -\mathbf{M}_1^{-1}(\mathbf{d}_1 + \mathbf{M}_2\mathbf{t}). \tag{10b}$$

Similarily, the solution of equation (9) is given by:

$$\mathbf{n} = -\mathbf{N}^{-1}\mathbf{g}. \tag{11}$$

Since all arrays in equations (10) and (11) are either 3×3 matrices or vectors of length 3, the solutions for ω, $\mathbf{t}$, and $\mathbf{n}$ can be determined easily.

5. Implementation

Let us consider the computations involved during one iteration. Using tensor notation (implicit summation over repeated indeces), we have:

$$\{\mathbf{M}_1\}_{i,j} = \iint_\Omega v_i v_j \, dx\, dy \quad \{\mathbf{M}_2\}_{i,j} = \left[\iint_\Omega r_k v_i s_j \, dx\, dy\right] n_k$$

$$\{\mathbf{M}_4\}_{i,j} = \left[\iint_\Omega r_k r_l s_i s_j \, dx\, dy\right] n_k n_l$$

$$\{\mathbf{N}\}_{i,j} = \left[\iint_\Omega s_k s_l r_i r_j \, dx\, dy\right] t_k t_l \tag{12}$$

$$\{\mathbf{d}_1\}_i = \iint_\Omega c v_i \, dx\, dy \quad \{\mathbf{d}_2\}_i = \left[\iint_\Omega c s_i r_j \, dx\, dy\right] n_j$$

$$\{\mathbf{g}\}_i = \left[\iint_\Omega c r_i s_j \, dx\, dy\right] t_j + \left[\iint_\Omega s_k r_i v_j \, dx\, dy\right] t_k w_j.$$

In the above equations, (cv_i), $(v_i v_j)$, $(cs_i r_j)$, $(r_k v_i s_j)$, and $(r_k r_l s_i s_j)$ depend only on x, y, E_x, E_y, and E_t, and so can be integrated over the image once. Therefore, the updating of the coefficients at each iteration only involves 27 multiplications to compute $\mathbf{M}_2$, 9 to compute $\mathbf{d}_2$, and 42 to compute each of $\mathbf{M}_4$, $\mathbf{N}$, and $\mathbf{g}$ (note that $\mathbf{M}_4$ and $\mathbf{N}$ are symmetric). This gives a total of 162 multiplications per iteration. Further, solving equations (10) and (11) for ω, $\mathbf{t}$, and $\mathbf{n}$ requires a total of 117 multiplications.

This iterative scheme has been implemented and tested for many cases. We will present a selected example in section 7.

6. Uniqueness

Our analytical as well as simulation results show (see [10] for proof) that there exists at most two solutions that generate the same optical flow (The existance but not necessarily the uniqueness of a dual solution has been shown in several papers [4,9,12,14]). The two solutions are related as follows:

$$\mathbf{n}' = k\mathbf{t}, \qquad \mathbf{t}' = k^{-1}\mathbf{n}, \quad \text{and} \quad \omega' = \omega + \mathbf{n} \times \mathbf{t}, \tag{13}$$

where k is any arbitrary constant chosen to scale the surface and translational motion parameters. Note that when the translational motion is perpendicular to the planar surface, a unique solution is obtained. Further, when the component of translational motion along the line of sight (Z-axis) is zero, the planar surface for the dual solution is parallel to the line of sight. In this case, the dual solution can be viewed as a degenerate one.

7. Experimental results

In the following example, we will demonstrate the sensitivity of the scheme to the initial condition. The image brightness function was generated using a multiplicative sinusoidal pattern (one that varies sinusoidally in both x and y directions), a 45° field of view was assumed, and the brightness gradients were computed analytically to avoid errors due to quantization and finite differencing of brightness values (In practice, the brightness values in two image frames are discretized first, and are then used to compute the brightness gradients using finite difference methods). Table 1 shows the results of two tests using different initial conditions. In each case, the algorithm converges to one of the two possible (true or dual) solutions. The results show that the error in each parameter after less than 30 iterations is within 10% of the true value.

In similar tests, with various motion and surface parameters, accurate results have been obtained in less than 40 iterations with a variety of initial conditions. More importantly, the algorithm eventually converged to one of the two possible solutions. The results have not been as satisfactory for the particular case where the translational motion component is (almost) perpendicular to the planar surface (The solution is unique in this case). In these cases, several hundred iterations were required to achieve reasonable accuracy. It appears that the behavior resembles that observed when the Newton-Raphson method is applied to a problem where two roots are very close to one another.

8. Summary

The problem of recovering the orientation of a planar surface and its motion from a sequence of images was investigated and formulated as one of unconstrained optimisation. Using conditions for optimality, the problem was reduced to solving a set of nine nonlinear algebraic equations and an implemented procedure based on an iterative scheme for solving these equations was presented. Through a selected example, it was shown that the algorithm could converge to either of two possible solutions (or the only solution when the translational motion vector is perpendicular to the surface). In practice, once a solution is obtained, the dual solution can be computed from equation (13). In several other cases tested, solutions with good accuracy have been obtained after 10-40 iterations.

0. Acknowledgments

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the Laboratory's Artificial Intelligence research is provided in part by the System Development Foundation.

Table 1. Test Results for a Selected Example

True Rotational Motion	A = .003	B = .001	C = -.01
True Translational Motion	U = .0005	V = -.005	W = .0125
True Surface Parameters	p = .2	q = .4	r = 1.
Dual Rotational Motion	A = .013	B = -.001	C = -.0112
Dual Translational Motion	U = .0025	V = .005	W = .0125
Dual Surface Parameters	p = .04	q = -.4	r = 1.

Initial Guess p = -.5 q = -1.5 r=1

Itr	A	B	C	U	V	W	p	q
10	.00531	.00260	-.01016	-.00069	-.00284	.01301	.35524	.1923
15	.00429	.00178	-.01008	-.00006	-.00384	.01291	.27623	.2742
20	.00353	.00137	-.01002	0.00024	-.00454	.01270	.23725	.3448
25	.00318	.00117	-.01000	0.00038	-.00485	.01257	.21718	.3814
30	.00305	.00107	-.01000	0.00045	-.00495	.01252	.20755	.3945
35	.00302	.00103	-.01000	0.00048	-.00499	.01250	.20323	.3984
40	.00300	.00101	-.01000	0.00049	-.00500	.01250	.20137	.3996
45	.00300	.00101	-.01000	0.00050	-.00500	.01250	.20058	.3999
50	.00300	.00100	-.01000	0.00050	-.00500	.01250	.20024	.4000
55	.00300	.00100	-.01000	0.00050	-.00500	.01250	.20010	.4000

Initial Guess p = -10(q= -5 r=1

Itr	A	B	C	U	V	W	p	q
10	.01302	-.00120	-.01118	.00266	.00503	.01247	.01941	-.4021
15	.01299	-.00108	-.01119	.00256	.00500	.01249	.03220	-.3902
20	.01299	-.00103	-.01120	.00253	.00500	.01250	.03692	-.3993
25	.01300	-.00101	-.01120	.00251	.00500	.01250	.03876	-.3996
30	.01300	-.00101	-.01120	.00250	.00500	.01250	.03950	-.3998
35	.01300	-.00100	-.01120	.00250	.00500	.01250	.03980	-.3099
40	.01300	-.00100	-.01120	.00250	.00500	.01250	.03992	-.4000
45	.01300	-.00100	-.01120	.00250	.00500	.01250	.03997	-.4000
50	.01300	-.00100	-.01120	.00250	.00500	.01250	.03909	-.4000

Note that without loss of generality. we have set $r = 1$, since the distance to the plane and the translational motion component can only be recovered up to a scale factor.

10. References

[I] Adiv, G. "Determining 3-D Motion and Structure from Optical Flow Generated by Several Moving Objects." COINS TR 84-07, Computer and Information Science, University of Massachusetts, Amherst. MA, April 1984.

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

[3] Fang, J.Q., Huang, T.S., "Solving Three Dimensional Small Rotation Motion Equations: Uniqueness, Algorithms, and Numerical Results," *Computer Vision, Graphics, and Image Processing,* Vol. 26, pp. 183 206, 1984.

[4] Hay, C.J., "Optical Motion and Space Perception, an Extension of Gibson's Analysis," *Psychological Review,* Vol. 73, pp. 550-565, 1966.

[5] Hildreth, E.G., *The Measurement of Visual Motion,* MIT Press, 1983.

|6j Horn, B.K.P., Schunck, B.C., "Determining Optical Flow," *Artificial Intelligence,* Vol. 17. pp. 185-203, 1981.

[7] Longuet-Higgins, H,C. "A Computer Algorithm for Reconstructing a Scene from Two Projections,'1 *Nature,* Vol. 293, pp. 131-133, 1981.

(8] Longuet-Higgins, H.C., Prazdny, K., "The Interpretation of a Moving Retinal Image," *Proceedings of the Royal Society of London,* Series D, Vol. 208, pp. 385 397, 1980.

[9] Maybank, S.J., "The Angular Velocity Associated with the Optical Flow Field Due to a Single Moving Rigid Plane," *Proceedings of the Sixth European Conference on Artificial Intelligence,* pp. 641 644, September 1984.

[10] Negahdaripour, S., Horn, D.K.P., "Direct Passive Navigation," A.I. Memo 821, MIT A.I. Lab, February 1985.

[II) Sugihara, K., Sugie, N., "Recovery of Rigid Structure from Orthographically Projected Optical Flow," TR 8304, Department of Information Science, Nagoya University, Nagoya, Japan, October 1983.

[12| Tsai. R.Y., Huang, T.S., Zhu, W.L., "Three-Dimensional Motion Parameters of a Rigid Planar Patch, 11: Singular Value Decomposition," *IEEE Trans, on Acoustics, Speech, and Signal Processing,* Vol. ASSP-30, No. 4, August 1982.

[13] Tsai, R.Y., Huang, T.S., "Uniqueness and Estimation of Three-Dimensional Motion Parameters of Rigid Objects with Curved Surfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-G, No. 1, January 1984.

[14] Waxman, A.M.. Ullman, S., "Surface Structure and 3-D Motion from Image Flow: A Kinematic Analysis," CAR-TR-24, Comp. Vis. Lab, Center for Automation Research, Univ. of Maryland. College Park. MD, October 1983.

[15] Waxman. A.M., Wohn, K.. "Contour Evolution, Neighborhood Deformation and Clobal Image Flow: Planar Surfaces in Motion." CAR-TR-58, Comp. Vis. Lab, Center for Automation Research, Univ. of Maryland, College Pnrk, MD, April 1984.

[1G| Wohn, K., Davis. L.S., Thrift, P., "Motion Estimation Based on Multiple Local Constraints and Nonlinear Smoothing," *Pattern Recognition,* Vol. 1G, No. G, pp. 5G3 570, 1983.