Direct Passive Navigation

SHAHRIAR NEGAHDARIPOUR AND BERTHOLD K. P. HORN

Abstract—**In this correspondence, we show how to recover the motion of an observer relative to a planar surface from image brightness derivatives. We do not compute the optical flow as an intermediate step, only the spatial and temporal brightness gradients (at a minimum of eight points). We first present two iterative schemes for solving nine nonlinear equations in terms of the motion and surface parameters that are derived from a least-squares fomulation. An initial pass over the relevant image region is used to accumulate a number of moments of the image brightness derivatives. All of the quantities used in the iteration are efficiently computed from these totals without the need to refer back to the image. We then show that either of two possible solutions can be obtained in closed form. We first solve a linear matrix equation for the elements of a 3 × 3 matrix. The eigenvalue decomposition of the symmetric part of the matrix is then used to compute the motion parameters and the plane orientation. A new compact notation allows us to show easily that there are at most two planar solutions.**

Index Terms—**Eigenvalue decomposition, least-squares, optical flow, planar surfaces, structure and motion.**

I. Introduction

The problem of recovering rigid body motion and surface structure from image sequences has been the topic of many research papers in the area of machine vision (the reader is referred to a survey of previous literature [1]). Two types of approaches, discrete and continuous, have been pursued. In the discrete approach, information about the displacements of a finite number of discrete points in the image is used to reconstruct the motion. 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 images. In the continuous approach, the optical flow, that is the apparent velocity of image brightness patterns, is used.

In much of the work on recovering surface structure and motion, it is assumed that either a correspondence between a sufficient number of feature points in successive frames has been established or that a reasonable estimate of the full optical flow field is available.

In general, identifying features involves determining gray-level corner points. For images of smooth objects, it is difficult to find good features or corners. Further, the correspondence problem has to be solved, that is, feature points from consecutive frames have to be matched.

The computation of the local flow field exploits a constraint equation between the local intensity changes and the two components of the optical flow. This only gives the component of flow in the direction of the intensity gradient. To compute the full flow field, one needs additional constraints such as the heuristic assumption that the flow field is locally smooth [4], [5]. This, in many cases, leads to an estimated optical flow field that is not the same as the true motion field.

In this corrspondence, we determine the motion of an observer relative to a planar surface directly from the image brightness derivatives without the need to compute the optical flow as an intermediate step. We restrict ourselves to planar surfaces since only three parameters are needed to specify the surface structure. We will first derive the image brightness constraint equation for the case of rigid body motion. A least squares formulation allows us to derive nine nonlinear equations, the so-called *planar motion field equations*, in terms of the motion and surface parameters. We present two iterative schemes for solving these equations. It is shown that all of the quantities used in the iteration can be computed efficiently from a number of moments of the image brightness derivatives that are accumulated through an initial pass of over the relevant image region. We therefore do not have to refer back to the image. We also show that a closed-form solution to the same problem can be obtained through a two-step procedure. We first solve a linear matrix equation for the elements of a 3 × 3 matrix equation using brightness derivatives (at a minimum of eight points). The eigenvalue decomposition of the symmetric part of this matrix allows us to compute the motion parameters and the plane orientation easily.

II. Preliminaries

We first recall some details about perspective projection, the motion field, the brightness change constraint equation, rigid body motion, and planar surfaces. This we do using vector notation in order to keep the resulting equations as compact as possible.

A. Perspective Projection

Let the center of projection be at the origin of a Cartesian coordinate system. Without loss of generality we assume that the effective focal length is unity. The image is formed on the plane $z = 1$, parallel to the xy-plane, that is, the optical axis lies along the z-axis. Let $\boldsymbol{R}$ be a point in the scene. Its projection in the image is $\boldsymbol{r}$, where

$$\boldsymbol{r} = \frac{1}{\boldsymbol{R} \cdot \hat{\boldsymbol{z}}} \boldsymbol{R}.$$

The z-component of $\boldsymbol{r}$ is clearly equal to one, that is $\boldsymbol{r} \cdot \hat{\boldsymbol{z}} = 1$.

B. Motion Field

The *motion field* is the vector field induced in the image plane by the relative motion of the observer with respect to the environ-

Manuscript received March 11, 1985; revised November 13, 1985 and April 30, 1986. Recommended for acceptance by W. B. Thompson. This correspondence describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology and the Robotics Group of the University of Hawaii. The M.I.T. artificial intelligence research was supported in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research Contract N00014-75-C-0643 and in part by the System Development Foundation. The University of Hawaii Robotics Group was supported in part by the National Science Foundation.

S. Negahdaripour is with the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

B. K. P. Horn is with the Department of Electrical Engineering, University of Hawaii, Honolulu, HI 96822, on leave from the Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, MA 02139.

IEEE Log Number 8609967.

Note: A paper describing the iterative solution to this problem was submitted by the authors on March 11, 1985 and accepted for publication on November 13, 1985. A second paper describing the closed form solution was submitted on November 18, 1985. At the suggestion of the reviewers and the Editor these two papers have been consolidated into one paper, which was accepted for publication on April 30, 1986.

0162-8828/87/0100-0168$01.00 © 1987 IEEE

ment. The *optical flow* is the apparent motion of brightness patterns. Under favorable circumstances the optical flow is identical to the motion field. The velocity of the image $\boldsymbol{r}$ of a point $\boldsymbol{R}$ is given by

$$\frac{d\boldsymbol{r}}{dt} = \frac{d}{dt}\frac{1}{\boldsymbol{R}\cdot\hat{\boldsymbol{z}}}\boldsymbol{R}.$$

For convenience, we introduce the notation $\boldsymbol{r}_t$ and $\boldsymbol{R}_t$ for the time derivatives of $\boldsymbol{r}$ and $\boldsymbol{R}$, respectively. We then have

$$\boldsymbol{r}_t = \frac{1}{\boldsymbol{R}\cdot\hat{\boldsymbol{z}}}\boldsymbol{R}_t - \frac{1}{(\boldsymbol{R}\cdot\hat{\boldsymbol{z}})^2}(\boldsymbol{R}_t\cdot\hat{\boldsymbol{z}})\boldsymbol{R},$$

which can also be written in the compact form

$$\boldsymbol{r}_t = \frac{1}{(\boldsymbol{R}\cdot\hat{\boldsymbol{z}})^2}(\hat{\boldsymbol{z}}\times(\boldsymbol{R}_t\times\boldsymbol{R})),$$

since $\boldsymbol{a}\times(\boldsymbol{b}\times\boldsymbol{c}) = (\boldsymbol{c}\cdot\boldsymbol{a})\boldsymbol{b} - (\boldsymbol{a}\cdot\boldsymbol{b})\boldsymbol{c}$. The vector $\boldsymbol{r}_t$ lies in the image plane, and so $(\boldsymbol{r}_t\cdot\hat{\boldsymbol{z}}) = 0$. Further, $\boldsymbol{r}_t = 0$, if $\boldsymbol{R}_t \| \boldsymbol{R}$, as expected.

Finally, noting that $\boldsymbol{R} = (\boldsymbol{R}\cdot\hat{\boldsymbol{z}})\boldsymbol{r}$, we get

$$\boldsymbol{r}_t = \frac{1}{\boldsymbol{R}\cdot\hat{\boldsymbol{z}}}(\hat{\boldsymbol{z}}\times(\boldsymbol{R}_t\times\boldsymbol{r})).$$

C. Rigid Body Motion

In the case of the observer moving relative to a rigid environment with translational velocity $\boldsymbol{t}$ and rotational velocity $\boldsymbol{\omega}$, we find that the motion of a point in the environment relative to the observer is given by

$$\boldsymbol{R}_t = -\boldsymbol{\omega}\times\boldsymbol{R} - \boldsymbol{t}.$$

Since $\boldsymbol{R} = (\boldsymbol{R}\cdot\hat{\boldsymbol{z}})\boldsymbol{r}$, we can write this as

$$\boldsymbol{R}_t = -(\boldsymbol{R}\cdot\hat{\boldsymbol{z}})\boldsymbol{\omega}\times\boldsymbol{r} - \boldsymbol{t}.$$

Substituting for $\boldsymbol{R}_t$ in the formula derived above for $\boldsymbol{r}_t$, we obtain

$$\boldsymbol{r}_t = -\left(\hat{\boldsymbol{z}}\times\left(\boldsymbol{r}\times\left(\boldsymbol{r}\times\boldsymbol{\omega} - \frac{1}{\boldsymbol{R}\cdot\hat{\boldsymbol{z}}}\boldsymbol{t}\right)\right)\right).$$

It is important to remember that there is an inherent ambiguity here, since the same motion field results when distance and the translational velocity are multiplied by an arbitrary constant. This can be seen easily from the above equation since the same image plane velocity is obtained if one multiplies both $\boldsymbol{R}$ and $\boldsymbol{t}$ by some constant.

D. Brightness Change Equation

The brightness of the image of a particular patch of a surface depends on many factors. It may for example vary with the orientation of the patch. In many cases, however, it remains at least aproximately constant as the surface moves in the environment. If we assume that the image brightness of a patch remains constant, we have

$$\frac{dE}{dt} = 0,$$

or

$$\frac{\partial E}{\partial \boldsymbol{r}}\cdot\frac{d\boldsymbol{r}}{dt} + \frac{\partial E}{\partial t} = 0,$$

where $\partial E/\partial \boldsymbol{r} = (\partial E/\partial x, \partial E/\partial y, 0)^T$ is the image brightness gradient. It is convenient to use the notation E_r for this quantity and E_t for the time derivative of the brightness. Then we can write the brightness change equation in the simple form

$$E_r\cdot\boldsymbol{r}_t + E_t = 0.$$

Substituting for $\boldsymbol{r}_t$ we get

$$E_t - E_r\cdot\left(\hat{\boldsymbol{z}}\times\left(\boldsymbol{r}\times\left(\boldsymbol{r}\times\boldsymbol{\omega} - \frac{1}{\boldsymbol{R}\cdot\hat{\boldsymbol{z}}}\boldsymbol{t}\right)\right)\right) = 0.$$

Now

$$E_r\cdot(\hat{\boldsymbol{z}}\times(\boldsymbol{r}\times\boldsymbol{t})) = (E_r\times\hat{\boldsymbol{z}})\cdot(\boldsymbol{r}\times\boldsymbol{t}) = ((E_r\times\hat{\boldsymbol{z}})\times\boldsymbol{r})\cdot\boldsymbol{t},$$

and by similar reasoning

$$E_r\cdot(\hat{\boldsymbol{z}}\times(\boldsymbol{r}\times(\boldsymbol{r}\times\boldsymbol{\omega}))) = (((E_r\times\hat{\boldsymbol{z}})\times\boldsymbol{r})\times\boldsymbol{r})\cdot\boldsymbol{\omega},$$

so we have

$$E_t - (((E_r\times\hat{\boldsymbol{z}})\times\boldsymbol{r})\times\boldsymbol{r})\cdot\boldsymbol{\omega} + \frac{1}{\boldsymbol{R}\cdot\hat{\boldsymbol{z}}}((E_r\times\hat{\boldsymbol{z}})\times\boldsymbol{r})\cdot\boldsymbol{t} = 0.$$

To make this constraint equation more compact, let us define $c = E_t$, $\boldsymbol{s} = (E_r\times\hat{\boldsymbol{z}})\times\boldsymbol{r}$, and $\boldsymbol{v} = -\boldsymbol{s}\times\boldsymbol{r}$; then, finally,

$$c + \boldsymbol{v}\cdot\boldsymbol{\omega} + \frac{1}{\boldsymbol{R}\cdot\hat{\boldsymbol{z}}}\boldsymbol{s}\cdot\boldsymbol{t} = 0.$$

This is the brightness change equation in the case of rigid body motion.

E. Planar Surface

A particularly impoverished scene is one consisting of a single planar surface. The equation for such a surface is

$$\boldsymbol{R}\cdot\boldsymbol{n} = 1,$$

where $\boldsymbol{n}/|\boldsymbol{n}|$ is a unit normal to the plane, and $1/|\boldsymbol{n}|$ is the perpendicular distance of the plane from the origin. Since $\boldsymbol{R} = (\boldsymbol{R}\cdot\hat{\boldsymbol{z}})\boldsymbol{r}$, we can write this as

$$\boldsymbol{r}\cdot\boldsymbol{n} = \frac{1}{\boldsymbol{R}\cdot\hat{\boldsymbol{z}}},$$

so the constraint equation becomes

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

This is the brightness change equation for a planar surface.

Note again the inherent ambiguity in the constraint equation. It is satisfied equally well by two planes with the same orientation but at different distances provided that the translational velocities are in the same proportions.

III. Recovering Motion and Structure

Given image brightness $E(x, y, t)$, and its spatial and time derivatives, E_r and E_t, over some region I in the image plane, we are to recover the translational and rotational motions, $\boldsymbol{t}$ and $\boldsymbol{\omega}$, as well as the plane $\boldsymbol{n}$. Using the constraint equation developed above, we could do this using image information at just a small number of points. At each point we get one constraint and we have nine unknowns to recover—or rather, eight, since we can recover the distance of the plane and the translational velocity only up to a scale factor. We will first present the iterative method. The motion parameters and the plane orientation are obtained from the solution of nine nonlinear equations derived from a least-squares formulation for minimizing the error in the brightness change constraint equation. We then present the closed-form solution to the same problem that involves a two-step procedure. First, we solve for nine intermediate parameters, the elements of a 3×3 matrix, using brightness derivatives at a minimum of eight points. We then solve for the motion parameters and the plane orientation from the eigenvalue decomposition of the symmetric part of this matrix.

A. Iterative Method: Least-Squares Formulation

Image brightness values are distorted with sensor noise and quantization error. These inaccuracies are further accentuated by methods used for estimating the brightness gradient. Thus it is not advisable to base a method on measurements at just a few points. Instead we propose to minimize the error in the brightness constraint equation over the whole region I in the image plane. So we wish to minimize

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

by suitable choice of the translational and rotational motion vectors $\boldsymbol{t}$ and $\boldsymbol{\omega}$, as well as the normal to the plane $\boldsymbol{n}$.

For an extremum of J we must have

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

That is,

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

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

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

These equations comprise nine nonlinear (scalar) algebraic equation in terms of the observer motion, $\boldsymbol{t}$ and $\boldsymbol{\omega}$, and the surface normal $\boldsymbol{n}$. We will call them the *planar motion field equations*. Some observations about these equations are in order. The first equation is linear in $\boldsymbol{\omega}$, $\boldsymbol{t}$, and $\boldsymbol{n}$. The second equation is linear in $\boldsymbol{\omega}$ and $\boldsymbol{t}$, but quadratic in $\boldsymbol{n}$. Finally, the last equation is linear in $\boldsymbol{\omega}$ and $\boldsymbol{n}$, but quadratic in $\boldsymbol{t}$. We will exploit the linearity of these equations to formulate two iterative schemes.

1) First Scheme: We can rearrange the planar motion field equations to get

$$\left[\iint_I (\boldsymbol{v}\boldsymbol{v}^T)\, dx\, dy\right]\boldsymbol{\omega} + \left[\iint_I (\boldsymbol{r}\cdot\boldsymbol{n})(\boldsymbol{v}\boldsymbol{s}^T)\, dx\, dy\right]\boldsymbol{t} = -\iint_I c\boldsymbol{v}\, dx\, dy,$$

$$\left[\iint_I (\boldsymbol{r}\cdot\boldsymbol{n})(\boldsymbol{s}\boldsymbol{v}^T)\, dx\, dy\right]\boldsymbol{\omega} + \left[\iint_I (\boldsymbol{r}\cdot\boldsymbol{n})^2(\boldsymbol{s}\boldsymbol{s}^T)\, dx\, dy\right]\boldsymbol{t} = -\iint_I c(\boldsymbol{r}\cdot\boldsymbol{n})\boldsymbol{s}\, dx\, dy,$$

$$\left[\iint_I (\boldsymbol{s}\cdot\boldsymbol{t})^2(\boldsymbol{r}\boldsymbol{r}^T)\, dx\, dy\right]\boldsymbol{n} = -\iint_I [c + (\boldsymbol{v}\cdot\boldsymbol{\omega})](\boldsymbol{s}\cdot\boldsymbol{t})\boldsymbol{r}\, dx\, dy.$$

The first pair can be grouped in the form

$$\begin{pmatrix} M_1 & M_2 \\ M_2^T & M_4 \end{pmatrix}\begin{pmatrix} \boldsymbol{\omega} \\ \boldsymbol{t} \end{pmatrix} = -\begin{pmatrix} \boldsymbol{d}_1 \\ \boldsymbol{d}_2 \end{pmatrix},$$

where

$$M_1 = \iint_I (\boldsymbol{v}\boldsymbol{v}^T)\, dx\, dy; \quad M_2 = \iint_I (\boldsymbol{r}\cdot\boldsymbol{n})(\boldsymbol{v}\boldsymbol{s}^T)\, dx\, dy,$$

$$M_4 = \iint_I (\boldsymbol{r}\cdot\boldsymbol{n})(\boldsymbol{s}\boldsymbol{s}^T)\, dx\, dy,$$

$$\boldsymbol{d}_1 = \iint_I c\boldsymbol{v}\, dx\, dy, \text{ and } \boldsymbol{d}_2 = \iint_I c(\boldsymbol{r}\cdot\boldsymbol{n})\boldsymbol{s}\, dx\, dy.$$

This can be solved for $\boldsymbol{t}$ and $\boldsymbol{\omega}$, given the surface normal $\boldsymbol{n}$. The last equation is

$$N_4\boldsymbol{n} = -\boldsymbol{g},$$

where

$$N_4 = \iint_I (\boldsymbol{s}\cdot\boldsymbol{t})^2(\boldsymbol{r}\boldsymbol{r}^T)\, dx\, dy,$$

$$\boldsymbol{g} = \iint_I [c + (\boldsymbol{v}\cdot\boldsymbol{\omega})](\boldsymbol{s}\cdot\boldsymbol{t})\boldsymbol{r}\, dx\, dy,$$

and can be solved for the surface normal $\boldsymbol{n}$, given the pair of vectors $\boldsymbol{t}$ and $\boldsymbol{\omega}$.

The motion vectors are given by

$$\boldsymbol{\omega} = (M_2^{-1}M_1 - M_4^{-1}M_2^T)^{-1}(M_4^{-1}\boldsymbol{d}_2 - M_2^{-1}\boldsymbol{d}_1),$$

$$\boldsymbol{t} = (M_1^{-1}M_2 - M_2^{-T}M_4)^{-1}(M_2^{-T}\boldsymbol{d}_2 - M_1^{-1}\boldsymbol{d}_1),$$

where $(^{-T})$ denotes the inverse of the transpose of a matrix. This can also be written in the form

$$\boldsymbol{t} = (M_4 - M_2^TM_1^{-1}M_2)^{-1}(M_2^TM_1^{-1}\boldsymbol{d}_1 - \boldsymbol{d}_2),$$

$$\boldsymbol{\omega} = -M_1^{-1}(\boldsymbol{d}_1 + M_2\boldsymbol{t}),$$

or

$$\boldsymbol{\omega} = (M_1 - M_2M_4^{-1}M_2^T)^{-1}(M_2M_4^{-1}\boldsymbol{d}_2 - \boldsymbol{d}_1),$$

$$\boldsymbol{t} = -M_4^{-1}(\boldsymbol{d}_2 + M_2^T\boldsymbol{\omega}).$$

The surface normal is simply given by

$$\boldsymbol{n} = -N_4^{-1}\boldsymbol{g}.$$

All arrays are either 3×3 matrices or vectors of length 3, and therefore, the solutions for $\boldsymbol{\omega}$, $\boldsymbol{t}$, and $\boldsymbol{n}$ can be computed easily. Actually, most of the indicated matrix inversions do not have to be carried out explicitly, since it is computationally cheaper to solve these linear matrix equations by elimination.

So, in summary, we start with an initial guess for $\boldsymbol{n}$. Using the above equations, we solve for $\boldsymbol{t}$ and $\boldsymbol{\omega}$ in terms of the current value of $\boldsymbol{n}$, and then for $\boldsymbol{n}$ in terms of the current values of $\boldsymbol{t}$ and $\boldsymbol{\omega}$. After this, we evaluate the improvement in the solution to either go to next iteration or stop if the solution has not improved.

2) Second Scheme: The first pair of the motion and surface recovery equations depend linearly on $\boldsymbol{t}$ and $\boldsymbol{\omega}$. As before,

$$\left[\iint_I (\boldsymbol{v}\boldsymbol{v}^T)\, dx\, dy\right]\boldsymbol{\omega} + \left[\iint_I (\boldsymbol{r}\cdot\boldsymbol{n})(\boldsymbol{v}\boldsymbol{s}^T)\, dx\, dy\right]\boldsymbol{t} = -\iint_I c\boldsymbol{v}\, dx\, dy,$$

$$\left[\iint_I (\boldsymbol{r}\cdot\boldsymbol{n})(\boldsymbol{s}\boldsymbol{v}^T)\, dx\, dy\right]\boldsymbol{\omega} + \left[\iint_I (\boldsymbol{r}\cdot\boldsymbol{n})^2(\boldsymbol{s}\boldsymbol{s}^T)\, dx\, dy\right]\boldsymbol{t} = -\iint_I c(\boldsymbol{r}\cdot\boldsymbol{n})\boldsymbol{s}\, dx\, dy,$$

which can be solved for $\boldsymbol{t}$ and $\boldsymbol{\omega}$ in terms of $\boldsymbol{n}$. Furthermore, the first and last equations depend linearly on $\boldsymbol{n}$ and $\boldsymbol{\omega}$:

$$\left[\iint_I (\boldsymbol{v}\boldsymbol{v}^T)\, dx\, dy\right]\boldsymbol{\omega} + \left[\iint_I (\boldsymbol{s}\cdot\boldsymbol{t})(\boldsymbol{v}\boldsymbol{r}^T)\, dx\, dy\right]\boldsymbol{n} = -\iint_I c\boldsymbol{v}\, dx\, dy,$$

$$\left[\iint_I (\boldsymbol{s}\cdot\boldsymbol{t})(\boldsymbol{r}\boldsymbol{v}^T)\, dx\, dy\right]\boldsymbol{\omega} + \left[\iint_I (\boldsymbol{s}\cdot\boldsymbol{t})^2(\boldsymbol{r}\boldsymbol{r}^T)\, dx\, dy\right]\boldsymbol{n} = -\iint_I c(\boldsymbol{s}\cdot\boldsymbol{t})\boldsymbol{r}\, dx\, dy.$$

Given $\boldsymbol{t}$, these may be solved for $\boldsymbol{n}$ and $\boldsymbol{\omega}$. For simplicity, let M_1, M_2, M_4, N_4, $\boldsymbol{d}_1$, and $\boldsymbol{d}_2$ be as defined earlier, and let:

$$N_1 = M_1, \quad \boldsymbol{d}_1 = \boldsymbol{e}_1,$$

$$N_2 = \iint_I (\boldsymbol{s}\cdot\boldsymbol{t})(\boldsymbol{v}\boldsymbol{r}^T)\, dx\, dy, \text{ and } \boldsymbol{e}_2 = \iint_I c(\boldsymbol{s}\cdot\boldsymbol{t})\boldsymbol{r}\, dx\, dy.$$

Then

$$\begin{pmatrix} M_1 & M_2 \\ M_2^T & M_4 \end{pmatrix}\begin{pmatrix} \omega \\ t \end{pmatrix} = -\begin{pmatrix} d_1 \\ d_2 \end{pmatrix},$$

and

$$\begin{pmatrix} N_1 & N_2 \\ N_2^T & N_4 \end{pmatrix}\begin{pmatrix} \omega \\ n \end{pmatrix} = -\begin{pmatrix} e_1 \\ e_2 \end{pmatrix}.$$

The solution of the above equations is given by

$$\omega = (M_2^{-1}M_1 - M_4^{-1}M_2^T)^{-1}(M_4^{-1}d_2 - M_2^{-1}d_1),$$

$$t = (M_1^{-1}M_2 - M_2^{-T}M_4)^{-1}(M_2^{-T}d_2 - M_1^{-1}d_1),$$

and

$$\omega = (N_2^{-1}N_1 - N_4^{-1}N_2^T)^{-1}(N_4^{-1}e_2 - N_2^{-1}e_1),$$

$$n = (N_1^{-1}N_2 - N_2^{-T}N_4)^{-1}(N_2^{-T}e_2 - N_1^{-1}e_1).$$

These may be rewritten in either of two asymmetrical forms shown earlier.

Again, most of the indicated matrix inversions do not have to be carried out explicitly, since we can solve the equations by elimination.

In this scheme, we start with an initial guess for n. We solve for t and ω in terms of the current value of n, and update t, then solve for n and ω in terms of the current value of t, and update n, and, finally, evaluate the improvement in the solution to either continue with the next iteration or stop if the solution has not improved.

3) Division of Labor: These methods would not be very attractive, if we had to perform integrations over the whole image region I during each iteration, in order to collect the matrices and vectors appearing in the equations. Fortunately, this is not necessary. One can see this by writing the equations for the components of the matrices and vectors using the summation convention of tensor calculus (that is, there is an implicit summation over any index that appears twice in an expression):

$$\{M_1\}_{ij} = \iint_I v_i v_j \, dx \, dy,$$

$$\{M_2\}_{ij} = \left| \iint_I v_i s_j r_k \, dx \, dy \right| n_k,$$

$$\{M_4\}_{ij} = \left| \iint_I s_i s_j r_k r_l \, dx \, dy \right| n_k n_l,$$

$$\{d_1\}_i = \iint_I c v_i \, dx \, dy, \quad \{d_2\}_i = \left| \iint_I c s_i r_j \, dx \, dy \right| n_j,$$

$$\{N_1\}_{ij} = \iint_I v_i v_j \, dx \, dy,$$

$$\{N_2\}_{ij} = \left| \iint_I v_i s_k r_j \, dx \, dy \right| t_k,$$

$$\{N_4\}_{ij} = \left| \iint_I s_k s_l r_i r_j \, dx \, dy \right| t_k t_l,$$

$$\{e_1\}_i = \iint_I c v_i \, dx \, dy, \quad \{e_2\}_i = \left| \iint_I c s_j r_i \, dx \, dy \right| t_j,$$

and

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

$M_1 = N_1$ and $d_1 = e_1$ do not depend on ω, t, or n, and so need only be computed once. Also, (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 r, E_r, and E_t, and so can be integrated over the image once. This appears to be a set of $3 + 9 + 9 + 27 + 81 = 129$ numbers, but, because of symmetry in $(v_i v_j)$, and $(r_k r_l s_i s_j)$, only 81 numbers have to be stored. These accumulated totals represent all the image information needed to solve the motion recovery problem.

In the first scheme, we only perform 279 multiplications per iteration; The updating of the coefficients of the planar motion field equations involves $27 + 9 + 42 + 42 + 42 = 162$ multiplications to compute M_2, d_2, M_4, N_4, and g (note that M_4 and N_4 are symmetric). The updating of ω, t, and n, in comparison, requires 117 multiplications.

In the second scheme, 696 multiplications are carried out at each iteration; we compute the matrices M_2, M_4 and the vector d_2, required for the first half of the iteration, in $27 + 42 + 9 = 78$ multiplications. The same number of multiplications is needed to compute the matrices N_2, N_4 and the vector e_2 required in the second half. Further, solving for ω and t takes about 270 multiplications, as does solving for ω and n in the second half of each iterative step.

Through a selected example, we will show that the second scheme has a much better convergence rate at the expense of more computation per iteration.

B. Uniqueness

It is important to establish whether more than one solution is possible. In general, this is clearly so, since an image of uniform brightness could correspond to an arbitrary uniform surface moving in an arbitrary way. So the brightness gradients, or lack of brightness gradients, can conspire to make the problem highly ambiguous. What we are interested in here is whether two different planar surfaces can give rise to the same motion field given two different translational and rotational motions of the imaging system.

In our terms then, the question becomes: given that the brightness change equation is satisfied for the motion t and ω and the planar surface n, is there another motion t' and ω' and another planar surface n' that satisfies the same equation at all points in the region I and for all possible ways of marking the surface? Note that we have to consider a whole image region, since the problem is underconstrained if we only have information along a line or at a point in the image. We also have to include the condition that the constraint should be satisfied for all possible surface markings to avoid the kind of ambiguity discussed above, where brightness gradients fortuitously line up with the motion field to create ambiguity.

1) Dual Solution: Suppose that two motions and two planar surfaces satisfy the brightness change equation. Then, we have

$$c + v \cdot \omega + (r \cdot n)(s \cdot t) = 0,$$

$$c + v \cdot \omega' + (r \cdot n')(s \cdot t') = 0.$$

Subtracting these equations, we get

$$v \cdot (\omega - \omega') + (r \cdot n)(s \cdot t) - (r \cdot n')(s \cdot t') = 0.$$

Now $v = -s \times r$, so

$$-(s \times r) \cdot (\omega - \omega') + (r \cdot n)(s \cdot t) - (r \cdot n')(s \cdot t') = 0,$$

or

$$-r \cdot ((\omega - \omega') \times s) + (r \cdot n)(s \cdot t) - (r \cdot n')(s \cdot t') = 0.$$

If we let $\omega = (\omega_1, \omega_2, \omega_3)^T$, then we can write

$$\omega \times s = \Omega s, \quad \text{where} \quad \Omega = \begin{pmatrix} 0 & -\omega_3 & +\omega_2 \\ +\omega_3 & 0 & -\omega_1 \\ -\omega_2 & +\omega_1 & 0 \end{pmatrix},$$

is a suitable (3×3) skew-symmetric matrix. The (i, j)th element of Ω equals $w_k \epsilon_{ijk}$, where ϵ_{ijk} is the permutation symbol. (It equals $+1$ when the ordered set i, j, k is obtained by an even permutation of the set 1, 2, and 3, it equals -1 when the ordered set is obtained by an odd permutation, and it is zero if two or more of the indexes are equal.)

Using this notation we can now write

$$-r^T(\Omega - \Omega')\, s + r^T(nt^T)\, s - r^T(n't'^T)s = 0,$$

or just

$$r^T[-(\Omega - \Omega') + nt^T - n't'^T]s = 0.$$

This is to be true for all points r in the image region I and all possible brightness gradients. So

$$-(\Omega - \Omega') + nt^T - n't'^T = \mathbf{0},$$

where the zero on the right-hand side here represents a 3×3 matrix of zeros. Now $\Omega^T = -\Omega$, since Ω is a skew-symmetric, so taking the transpose of the equation we get

$$+(\Omega - \Omega') + tn^T - t'n'^T = \mathbf{0}.$$

Adding the two equations allows us to eliminate $(\Omega - \Omega')$, and we end up with

$$(nt^T + tn^T) = (n't'^T + t'n'^T).$$

The trace (sum of the diagonal elements) of nt^T is just $(n \cdot t)$, so we see immediately that $(n \cdot t) = (n' \cdot t'^T)$. But the above matrix equation involving the dyadic products of n and t as well as of n' and t' is much more constraining.

Consider the following three possibilities:

1) If $|n'| = 0$ or $|t'| = 0$, then their dyadic product is a 3×3 matrix of zeros. In this case the above equation is satisfied if and only if $|n| = 0$ or $|t| = 0$.

2) If $n' \| n$, $t' \| t$ and $|n||t'| = |n||t|$, then the two sums of dyadic products are equal and the above equation is satisfied.

3) If $n' \| t$, $t' \| n$ and $|n'||t'| = |n||t|$, then the two sums of dyadic products are also equal and the above equation is satisfied.

It turns out that there are no other ways to satisfy the equation. This can be shown using elementary properties of dyadic products (see [8]) or by inspection of the six components of the above equation (because of symmetry there are only six independent components).

The first case above corresponds to purely rotational motion, because either the translational motion is zero, or the planar surface is infinitely far away, and the translation does not generate a perceptible component of the motion field. The solution is unique in this case, because we find $(\Omega - \Omega') = \mathbf{0}$, when we substitute back into the matrix equation. (This is nothing new, since it has been known for some time that the solution is unique in the case of purely rotational and purely translational motion [2].)

In the second case we find that $nt^T = n't'^T$, since the vectors are parallel and the product of their size is constrained by the condition $n \cdot t = n' \cdot t'$, derived earlier. Thus once again $(\Omega - \Omega') = \mathbf{0}$. Nothing new is obtained here, since we already know that we can change the lengths of the vectors n and t as long as the product of their lengths remains constant.

The third case is the most interesting. Here we have $tn^T = n't'^T$ so that

$$-(\Omega - \Omega') + (nt^T - tn^T) = \mathbf{0},$$

and thus

$$-(\Omega - \Omega')\, x + (nt^T - tn^T)x = \mathbf{0},$$

for an arbitrary vector x. That is,

$$x \times (\omega - \omega') + x \times (n \times t) = \mathbf{0},$$

for an arbitrary vector x, so that

$$\omega - \omega' + n \times t = \mathbf{0},$$

or $\omega' = \omega + n \times t$. To summarize then, if we ignore scaling of the normal and the translational velocity, we obtain a dual solution,

TABLE I

THE TRUE MOTION AND SURFACE PARAMETERS, AND A SUMMARY OF THE RESULTS OF A SIMULATION THAT CONVERGES TO THE TRUE SOLUTION USING THE FIRST SCHEME

True Rotational Motion Parameters	$w_1 = .003$	$w_2 = .001$	$w_3 = -.01$
True Translational Motion Parameters	$t_1 = .0005$	$t_2 = -.005$	$t_3 = .0125$
True Parameters of the Surface	$n_1 = .2$	$n_2 = .4$	$n_3 = 1.0$
Initial Guess for the Simulation	$n_1 = 100.$	$n_2 = 5.$	$n_3 = -1.$

Iter. No.	(Rotational Par's) w_1	w_2	w_3	(Translational Par's) t_1	t_2	t_3	(Surface Par's) n_1	n_2	n_3
10	.00531	.00260	-.01016	-.00069	-.00284	.01301	.35524	.1923	1.
15	.00429	.00178	-.01008	-.00006	-.00384	.01291	.27623	.2742	1.
20	.00353	.00137	-.01002	.00024	-.00454	.01270	.23725	.3448	1.
25	.00318	.00117	-.01000	.00038	-.00485	.01257	.21718	.3814	1.
30	.00305	.00107	-.01000	.00045	-.00495	.01252	.20755	.3945	1.
35	.00302	.00103	-.01000	.00048	-.00499	.01250	.20323	.3984	1.
40	.00300	.00101	-.01000	.00049	-.00500	.01250	.20137	.3996	1.
45	.00300	.00101	-.01000	.00050	-.00500	.01250	.20058	.3999	1.
50	.00300	.00100	-.01000	.00050	-.00500	.01250	.20024	.4000	1.
55	.00300	.00100	-.01000	.00050	-.00500	.01250	.20010	.4000	1.
60	.00300	.00100	-.01000	.00050	-.00500	.01250	.20004	.4000	1.
65	.00300	.00100	-.01000	.00050	-.00500	.01250	.20002	.4000	1.
70	.00300	.00100	-.01000	.00050	-.00500	.01250	.20000	.4000	1.

given by

$$n' = t,\ t' = n \quad \text{and} \quad \omega' = \omega + n \times t.$$

Hay was the first to show the existence of the dual solution [3], although the result has apparently been independently rediscovered several times since then [6], [7], [9]. (The most recent papers [6], [7] came to our attention only after completion of our version of the proof.)

This dual solution is not different from the original one in the special case that the motion is perpendicular to the planar surface, that is, $n \| t$. In this case the solution is unique. Further, if $t \cdot \hat{z} = 0$, then $n' \cdot \hat{z} = 0$. This corresponds to a planar surface parallel to the observer's line of sight, and may be considered to be a degenerate case.

C. A Selected Example

We now present the results of a simulation. It is noteworthy to mention that in all simulations performed, our algorithms have converged to a solution. However, the number of iterations for convergence to a solution depends on the initial condition (as is the case with all iterative schemes developed for solving nonlinear equations). In this example, we will demonstrate the sensitivity of both schemes 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 image brightness gradients were computed analytically to avoid errors due to image brightness quantization and finite difference approximations of the brightness gradient. In practice, the brightness at image points in two frames would be discretized first, and the gradient computed using finite difference methods.

Table I shows the true motion and surface parameters, and the results of a simulation that converged to the true solution using the first scheme described earlier. In Table II, the dual solution for the true motion and surface parameters, and the results of a simulation that converged to the dual solution are tabulated. In both cases, the solution after various number of iterations are given. The results show that in the first case, the error in each parameter after less than 30 iterations is within 10 percent of the exact value. In the second case, this accuracy is achieved in less than 20 iterations. Similar results are presented in Tables III and IV for the second scheme. Here, very good accuracy is achieved in less than 10 iterations for the true solution and about 5 iterations for the dual solution.

TABLE II
THE DUAL MOTION AND SURFACE PARAMETERS, AND A SUMMARY OF THE RESULTS OF A SIMULATION THAT CONVERGES TO THE DUAL SOLUTION USING THE FIRST SCHEME

Dual Rotational Motion Parameters	$w_1 = .013$	$w_2 = -.001$	$w_3 = -.0112$
Dual Translational Motion Parameters	$t_1 = .0025$	$t_2 = .005$	$t_3 = .0125$
Parameters of the Dual Surface	$n_1 = .04$	$n_2 = -.4$	$n_3 = 1.0$
Initial Guess for the Simulation	$n_1 = .5$	$n_2 = 1.5$	$n_3 = -1.$

Iter.	(Rotational Par's)			(Translational Par's)			(Surface Par's)		
No.	w_1	w_2	w_3	t_1	t_2	t_3	n_1	n_2	n_3
10	.01302	-.00120	-.01118	.00266	.00503	.01247	.01941	-.4021	1.
15	.01299	-.00108	-.01119	.00256	.00500	.01249	.03220	-.3992	1.
20	.01299	-.00103	-.01120	.00253	.00500	.01250	.03692	-.3993	1.
25	.01300	-.00101	-.01120	.00251	.00500	.01250	.03876	-.3996	1.
30	.01300	-.00101	-.01120	.00250	.00500	.01250	.03950	-.3998	1.
35	.01300	-.00100	-.01120	.00250	.00500	.01250	.03980	-.3999	1.
40	.01300	-.00100	-.01120	.00250	.00500	.01250	.03992	-.4000	1.
45	.01300	-.00100	-.01120	.00250	.00500	.01250	.03997	-.4000	1.
50	.01300	-.00100	-.01120	.00250	.00500	.01250	.03999	-.4000	1.

TABLE III
THE TRUE MOTION AND SURFACE PARAMETERS, AND A SUMMARY OF THE RESULTS OF A SIMULATION THAT CONVERGES TO THE TRUE SOLUTION USING THE SECOND SCHEME

True Rotational Motion Parameters	$w_1 = .003$	$w_2 = .001$	$w_3 = -.01$
True Translational Motion Parameters	$t_1 = .0005$	$t_2 = -.005$	$t_3 = .0125$
True Parameters of the Surface	$n_1 = .2$	$n_2 = .4$	$n_3 = 1.0$
Initial Guess for the Simulation	$n_1 = 100.$	$n_2 = 5.$	$n_3 = -1.$

Iter.	(Rotational Par's)			(Translational Par's)			(Surface Par's)		
No.	w_1	w_2	w_3	t_1	t_2	t_3	n_1	n_2	n_3
5	.00254	.00105	-.00990	.00039	-.00546	.01202	.20581	.44259	1.
10	.00296	.00100	-.00999	.00049	-.00504	.01246	.20039	.40375	1.
15	.00300	.00100	-.01000	.00050	-.00500	.01250	.20004	.40037	1.
20	.00300	.00100	-.01000	.00050	-.00500	.01250	.20000	.40004	1.
25	.00300	.00100	-.01000	.00050	-.00500	.01250	.20000	.40000	1.

TABLE IV
THE DUAL MOTION AND SURFACE PARAMETERS, AND A SUMMARY OF THE RESULTS OF A SIMULATION THAT CONVERGES TO THE DUAL SOLUTION USING THE SECOND SCHEME

Dual Rotational Motion Parameters	$w_1 = .013$	$w_2 = -.001$	$w_3 = -.0112$
Dual Translational Motion Parameters	$t_1 = .0025$	$t_2 = .005$	$t_3 = .0125$
Parameters of the Dual Surface	$n_1 = .04$	$n_2 = -.4$	$n_3 = 1.0$
Initial Guess for the Simulation	$n_1 = .5$	$n_2 = 1.5$	$n_3 = -1.$

Iter.	(Rotational Par's)			(Translational Par's)			(Surface Par's)		
No.	w_1	w_2	w_3	t_1	t_2	t_3	n_1	n_2	n_3
5	.01330	-.00102	-.01122	.00248	.00531	.01221	.03790	-.42663	1.
10	.01303	-.00100	-.01120	.00250	.00503	.01248	.03979	-.40245	1.
15	.01300	-.00100	-.01120	.00250	.00500	.01250	.03998	-.40024	1.
20	.01300	-.00100	-.01120	.00250	.00500	.01250	.04000	-.40002	1.
25	.01300	-.00100	-.01120	.00250	.00500	.01250	.04000	-.40000	1.

In similar tests, with various motion and surface parameters, accurate results have been obtained in less than 40 iterations using the first scheme and a variety of initial conditions. The same accuracy for the second scheme required less than 15 iterations. More importantly, both schemes eventually converged to one of the two possible solutions. However, the results for the particular case where the translational motion vector is (almost) parallel to the surface normal have not been as satisfactory. In these cases, several hundred iterations were required to achieve reasonable accuracy, even with the second scheme. Although the nature of this behavior has not been investigated in detail, it appears to resemble that observed when the Newton–Raphson method is applied to a problem where two roots are very close to one another.

D. Closed-Form Solution: Essential Parameters for Planar Surfaces

The brightness change equation can be written as

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

Using the identify $(\mathbf{r} \times \mathbf{s}) \cdot \boldsymbol{\omega} = \mathbf{r} \cdot (\mathbf{s} \times \boldsymbol{\omega})$, we obtain

$$c - \mathbf{r} \cdot (\boldsymbol{\omega} \times \mathbf{s}) + (\mathbf{r} \cdot \mathbf{n})(\mathbf{s} \cdot \mathbf{t}) = 0.$$

We now use the isomorphism between vectors and skewsymmetric matrices. Let us define

$$\boldsymbol{\Omega} = \begin{pmatrix} 0 & -\omega_3 & +\omega_2 \\ +\omega_3 & 0 & -\omega_1 \\ -\omega_2 & +\omega_1 & 0 \end{pmatrix},$$

then, $\boldsymbol{\Omega}\mathbf{s} = (\boldsymbol{\omega} \times \mathbf{s})$, and we conclude that

$$c - \mathbf{r}^T\boldsymbol{\Omega}\mathbf{s} + (\mathbf{r}^T\mathbf{n})(\mathbf{t}^T\mathbf{s}) = 0,$$

or

$$c + \mathbf{r}^T(-\boldsymbol{\Omega} + \mathbf{n}\mathbf{t}^T)\mathbf{s} = 0.$$

If we define

$$\mathbf{P} = \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & p_9 \end{pmatrix} = -\boldsymbol{\Omega} + \mathbf{n}\mathbf{t}^T,$$

we can finally write

$$c + \mathbf{r}^T\mathbf{P}\mathbf{s} = 0.$$

We will refer to $\{p_i\}$ as the *essential parameters* (in agreement with Tsai and Huang [10]) since these parameters contain all the information about the planar surface and motion parameters. The above constraint equation is linear in the elements of $\mathbf{P}$. Several such equations, for different image points, can be used to solve for these parameters. We will show how the special structure of $\mathbf{P}$ can be exploited to recover the motion and plane parameters very easily.

Note that the essential parameters are not independent. This is because $\mathbf{P}$ is not an arbitrary 3×3 matrix. It has a special structure as a result of the fact that it is the sum of a skew-symmetric matrix and a dyadic product. It takes three parameters to specify $\boldsymbol{\omega}$ (and hence $\boldsymbol{\Omega}$), three to specify $\mathbf{n}$, and another three for $\mathbf{t}$. The matrix $\mathbf{P}$, however, is unchanged if we replace $\mathbf{n}$ by $k\mathbf{n}$ and $\mathbf{t}$ by $(1/k)\mathbf{t}$ for any nonzero k. Thus, there are actually only eight degrees of freedom, not nine.

Equivalently, we can say that there is one constraint on $\mathbf{P}$. Since $\boldsymbol{\Omega}^T = -\boldsymbol{\Omega}$, it follows that

$$\mathbf{P}^* = \mathbf{P} + \mathbf{P}^T = \mathbf{n}\mathbf{t}^T + \mathbf{t}\mathbf{n}^T.$$

A dyadic product has rank one, or less. The sum of two dyadic products has at most rank two. So we conclude that

$$\det(\mathbf{P} + \mathbf{P}^T) = 0.$$

This constraint can be expressed in terms of the essential parameters as

$$p_1(p_5p_9 - p_6p_8) + p_4(p_3p_8 - p_2p_9) + p_7(p_2p_6 - p_3p_5) = 0.$$

We can use this equation, for example, to solve for p_9 given p_1, p_2, $\cdots$, p_8. It is difficult to use this equation directly when one attempts to find $\mathbf{P}$ from image brightness measurements.

There is a simple way around this problem, however. Note that $r^T s = 0$, because $s = ((E_r \times \hat{z}) \times r)$. So $r^T I s = 0$, and

$$c + r^T(P + lI)s = 0,$$

for arbitrary l. If we let $P' = P + lI$, we can write

$$c + r^T P' s = 0,$$

and conclude that we cannot recover P from image brightness measurements alone. To find P, we must impose the constraint $\det(P + P^T) = 0$. To avoid dealing directly with the resulting nonlinear relation between the essential parameters, we first find any P' that satisfies the above brightness change constraint equation for all image points being considered, and then determine l such that $P = P' - lI$ satisfies

$$\det(P + P^T) = 0.$$

Now,

$$\det(P + P^T) = \det(P' + P'^T - 2lI) = 0,$$

so that $2l$ must be an eigenvalue of the real symmetric matrix

$$P'^* = P' + P'^T.$$

It will become apparent, in the next section, that we ought to choose the middle one of the three real eigenvalues of P'^* for $2l$.

In summary, the overall plan is to find any matrix P' that satisfies the image brightness constraint equation,

$$c + r^T P' s = 0,$$

at a suitable number of image points and consequently determine P'^*. We can then solve for the middle eigenvalue of P'^* (which is $2l$) so as to construct the singular matrix $P = P' - lI$, and from that we finally determine n and t as well as Ω (and hence ω) using the relationship

$$P = -\Omega + nt^T.$$

1) Recovering Essential Parameters: We are looking for a matrix P' that satisfies the brightness change equation,

$$c + r^T P' s = 0,$$

at a chosen number of image points. Now,

$$r^T P' s = \text{Trace}\ \{(sr^T)P'\},$$

or

$$r^T P' s = \text{Flat}\ (sr^T) \cdot \text{Flat}\ (P'),$$

where Flat (M) is the vector obtained from the matrix M by adjoining its rows. So we can write the brightness change equation in the form

$$c + a^T p' = 0,$$

where

$$p' = (p_1', p_2', \cdots, p_9')^T,$$

$$a = (r_1 s_1, r_1 s_2, r_1 s_3, r_2 s_1, r_2 s_2, r_2 s_3, r_3 s_1, r_3 s_2, r_3 s_3)^T.$$

We first consider finding p' from the image brightness derivatives at the minimum number of points necessary. Later, we consider instead a least-squares procedure that takes into account information in a whole image region.

From the derivatives of the brightness at the ith image point considered, we can construct the vector a_i such that

$$a_i^T p' = -c_i.$$

As discussed above, there are really only eight independent degrees of freedom. So we can arbitrarily fix one of the components of the vector p'. This means that we can solve for the other eight using constraint equations derived from eight image points.

Let $p' = (p_1', p_2', \cdots, p_8', 0)^T$ denote the solution obtained by setting the last element equal to zero. If we define

$$\tilde{p}' = (p_1', p_2', \cdots, p_8')^T,$$

$$\tilde{a} = (r_1 s_1, r_1 s_2, r_1 s_3, r_2 s_1, r_2 s_2, r_2 s_3, r_3 s_1, r_3 s_2)^T,$$

then the above constraint equation reduces to

$$\tilde{a}_i^T \tilde{p}' = -c_i.$$

Using eight independent points, we can solve the following linear matrix equation:

$$A\tilde{p}' = -c,$$

where

$$A = (\tilde{a}_1, \tilde{a}_2, \cdots, \tilde{a}_8)^T, \quad c = (c_1, c_2, \cdots, c_8)^T.$$

The solution of the above equation is

$$\tilde{p}' = -A^{-1}c.$$

Image intensity values are corrupted with sensor noise and quantization. These inaccuracies are further accentuated by methods used for estimating the brightness gradient. Thus it is not advisable to base a method on measurements at just a few points. Instead we propose to minimize the error in the brightness constraint equation over the whole region I in the image plane. So we choose the vector $\tilde{p}'$ that minimizes

$$\iint_I (\tilde{a}^T \tilde{p}' + c)^2\, dx\, dy.$$

The solution, in this case, is given by

$$\tilde{p}' = -\left(\iint_I \tilde{a}\tilde{a}^T\, dx\, dy\right)^{-1} \left(\iint_I c\tilde{a}\, dx\, dy\right).$$

In either case, we construct p' by adjoining a zero to the vector $\tilde{p}'$. The result immediately gives us the matrix P'. We determine the eigenvalues of P'^* so that we can construct P^* by subtracting the identity matrix times twice the middle eigenvalue from P'^*. We can also determine P by subtracting the identity matrix times the middle eigenvalue from P'. At this point, we are ready to recover t, ω, and n.

Note that we do not have to repeat the eigenvalue–eigenvector analysis, since P^* has the same eigenvectors as P'^*, and its eigenvalues are merely shifted so as to make the middle one equal to zero. This follows from the fact that if u and λ are an eigenvector–eigenvalue pair of P'^*, that is,

$$P'^* u = \lambda u,$$

then u and $(\lambda - 2l)$ are an eigenvector–eigenvalue pair of P^*, since

$$P^* u = (P'^* - 2lI)\, u = (\lambda - 2l)u.$$

2) Recovering Motion and Structure: We now show how to compute the parameters of the translational motion and the plane orientation from the essential parameters. When we have done this, we will be able to also find the rotational parameters using

$$\Omega = nt^T - P.$$

As we saw before

$$P^* = P + P^T = tn^T + nt^T,$$

since Ω is skew-symmetric. Let us use the notation $\sigma = |n||t|$, and $\tau = \hat{n} \cdot \hat{t}$, where

$$\hat{n} = \frac{n}{|n|}, \quad \text{and} \quad \hat{t} = \frac{t}{|t|},$$

are the unit vectors in the directions of the surface normal and the translation vector, respectively. Then,

$$\text{Trace}\ (P^*) = \text{Trace}\ (P) + \text{Trace}\ (P^T) = 2n \cdot t = 2\sigma\tau.$$

It turns out that $\hat{n}$ and $\hat{t}$ can be easily recovered from the eigenvectors of the matrix P^*. In the following lemma, we show that the eigenvectors of P^* are combinations of the sought after vectors $\hat{n}$ and $\hat{t}$.

Lemma 1: Let $P^* = U\Lambda U^T$ be the eigenvalue decomposition of $P^* = (tn^T + nt^T)$. If n is not parallel to t, then,

$$\Lambda = \text{Diag}\,(\sigma(\tau - 1), 0, \sigma(\tau + 1)),$$

and,

$$U = \left[\frac{\hat{t} - \hat{n}}{\sqrt{2(1-\tau)}}\ \frac{\hat{t} \times \hat{n}}{\sqrt{1-\tau^2}}\ \frac{\hat{t} + \hat{n}}{\sqrt{2(1+\tau)}}\right].$$

Proof: Note that

$$P^* = \sigma(\hat{t}\hat{n}^T + \hat{n}\hat{t}^T).$$

Now $(\hat{t} \times \hat{n})$ is the eigenvector with eigenvalue zero since

$$P^*(\hat{t} \times \hat{n}) = \sigma(\hat{t}\hat{n}^T + \hat{n}\hat{t}^T)(\hat{t} \times \hat{n})$$
$$= \sigma\hat{t}[\hat{n}\hat{t}\hat{n}] + \sigma\hat{n}[\hat{t}\hat{n}\hat{t}] = \mathbf{0}.$$

Since P^* is real symmetric, it has three orthogonal eigenvectors. The other two eigenvectors must, therefore, be in the plane containing $\hat{t}$ and $\hat{n}$. Let $u = \alpha\hat{t} + \beta\hat{n}$ and λ denote an eigenvector–eigenvalue pair for some α and β (to be determined). Then,

$$\sigma(\hat{t}\hat{n}^T + \hat{n}\hat{t}^T)(\alpha\hat{t} + \beta\hat{n}) = \lambda(\alpha\hat{t} + \beta\hat{n}),$$

that becomes

$$\sigma[\alpha(\hat{t}\cdot\hat{n}) + \beta(\hat{n}\cdot\hat{n})]\hat{t} + \sigma[\alpha(\hat{t}\cdot\hat{t}) + \beta(\hat{t}\cdot\hat{n})]\hat{n} = \lambda\alpha\hat{t} + \lambda\beta\hat{n}.$$

Since $(\hat{t}\cdot\hat{n}) = \tau$, we can write

$$\begin{pmatrix}\sigma\tau - \lambda & \sigma \\ \sigma & \sigma\tau - \lambda\end{pmatrix}\begin{pmatrix}\alpha \\ \beta\end{pmatrix} = \begin{pmatrix}0 \\ 0\end{pmatrix}.$$

For this pair of homogeneous equations to have a nontrivial solution for α and β, the determinant of the 2×2 coefficient matrix must be zero, that is,

$$(\sigma\tau - \lambda)^2 - \sigma^2 = 0,$$

or

$$\lambda = \sigma(\tau \pm 1).$$

Substituting for λ into the earlier equations, we obtain

$$\alpha = \pm\beta.$$

Note that $\sigma(\tau - 1) < 0$ and $\sigma(\tau + 1) > 0$ because $\tau < 1$, as it is the cosine of the angle between $\hat{n}$ and $\hat{t}$. So one eigenvalue is negative and one is positive. (This is why we choose to make the middle eigenvalue zero when constructing P^* from P'^*.) We find that eigenvectors corresponding to the eigenvalues $\lambda_1 = \sigma(\tau - 1)$ and $\lambda_3 = \sigma(\tau + 1)$ are $\hat{t} - \hat{n}$ and $\hat{t} + \hat{n}$, respectively. If we normalize these, we obtain the unit vectors

$$u_1 = \frac{\hat{t} - \hat{n}}{\sqrt{2(1-\tau)}} \text{ and } u_3 = \frac{\hat{t} + \hat{n}}{\sqrt{2(1+\tau)}}.$$

Note that we can determine $\sigma = |n||t|$ from

$$\sigma = \tfrac{1}{2}(\lambda_3 - \lambda_1).$$

■

The equations for u_1 and u_3 are linear in $\hat{t}$ and $\hat{n}$, and so can be easily solved for these vectors:

$$\hat{n} = \sqrt{\tfrac{1}{2}(1+\tau)}\,u_3 - \sqrt{\tfrac{1}{2}(1-\tau)}\,u_1,$$
$$\hat{t} = \sqrt{\tfrac{1}{2}(1+\tau)}\,u_3 + \sqrt{\tfrac{1}{2}(1-\tau)}\,u_1.$$

The sign of the eigenvectors are arbitrary. If we change the sign of u_1, we obtain instead

$$\hat{n} = \sqrt{\tfrac{1}{2}(1+\tau)}\,u_3 + \sqrt{\tfrac{1}{2}(1-\tau)}\,u_1,$$
$$\hat{t} = \sqrt{\tfrac{1}{2}(1+\tau)}\,u_3 - \sqrt{\tfrac{1}{2}(1-\tau)}\,u_1,$$

where $\hat{n}$ and $\hat{t}$ are interchanged. This is the dual solution.

The signs of the two eigenvectors can be chosen independently. This might suggest that there are a total of four different solutions for $\hat{n}$ and $\hat{t}$. We show next that two of these solutions can be discarded because they correspond to viewing the planar surface "from behind." We assume that the visible part of the plane is the bounding surface of some solid object. We chose to define the orientation of the surface using the inward pointing normal n. The equation of the plane is $R \cdot n = 1$, or $(r \cdot n)(R \cdot \hat{z}) = 1$, since

$$R = (R \cdot \hat{z})r.$$

Now, $R \cdot \hat{z} = Z$ is positive for points in front of the viewer, and so $r \cdot n$ must be positive for points on the visible portion of the plane. The equation $r \cdot n = 0$ corresponds to a line in the image. Points on one side of this line, for which $r \cdot n > 0$, can be images of points on the plane defined by the inward pointing normal n. Conversely, points on the other side of the line, where $r \cdot n < 0$, cannot. They can be thought of as images of points on a parallel but oppositely oriented plane corresponding to the vector $-n$. We are analyzing brightness gradients for a particular image region. If $r \cdot n > 0$ for points in this region, then n is a possible solution for the surface normal. If $r \cdot n < 0$ for points in this region, then $-n$ is a possible solution. If $r \cdot n > 0$ for some points and $r \cdot n < 0$ for others, then we are not dealing with the image of a single planar surface.

Also, note that we can recover t and n up to a scale factor. We can let t to be a unit vector without loss of generality. Then, n can be found as follows:

$$n = |n|\hat{n} = |n||t|\hat{n} = \sigma n,$$

using the known value of σ.

So far, we have assumed that n and t are not parallel. In the special case that $\hat{t} \| \hat{n}$, we have

$$P^* = \sigma(\hat{t}\hat{n}^T + \hat{n}\hat{t}^T) = 2\sigma\hat{n}\hat{n}^T.$$

This dyadic product has rank one, that is, it only has one nonzero eigenvalue. This is easy to show since any vector perpendicular to $\hat{n}$ is an eigenvector with zero eigenvalue. Also, $\hat{n}$ is an eigenvector with eigenvalue 2σ.

So if we find that P'^* has two equal eigenvalues (that is P^* has two zero eigenvalues), then we conclude that $\hat{n}$ and $\hat{t}$ are parallel and equal to the eigenvector corresponding to the remaining eigenvalue.

We then solve for the rotation parameters by substituting the solutions for n and t into the equation

$$\Omega = nt^T - P.$$

Even though we gave a complete and compact proof of the dual solution earlier, it is intriguing to confirm those results with our closed-form solution. We showed that the two solutions are related by

$$n' = |n|t,\ t' = \frac{n}{|n|},\ \omega' = \omega + n \times t,$$

where we have arbitrarily set $|t| = 1$. The two solutions given earlier for n and t already satisfy the duality relationship given above. The identity

$$(nt^T - tn^T)\,x = x \times (n \times t),$$

holds for any vector x. Using this in

$$\omega' \times x = (\omega + n \times t) \times x = \omega \times x + (n \times t) \times x,$$

we arrive at

$$\omega' \times x = \omega \times x - (nt^T - tn^T)x,$$

or

$$\Omega' x = (\Omega - n t^T + t n^T) x.$$

If this is to be true for all vectors x, we must have

$$\Omega' = \Omega - n t^T + t n^T.$$

So, we finally obtain

$$-\Omega' + n' t'^T = -\Omega + n t^T - t n^T + t n^T,$$

or,

$$-\Omega' + n' t' = -\Omega + n t^T = P.$$

We conclude that n', t', and ω', as defined above, constitute a second solution since they lead to the same set of essential parameters.

IV. Summary

The problem of recovering the motion of an observer relative to a planar surface directly from the changing images (direct passive navigation) was investigated and two solution procedures were presented.

We first formulated an unconstrained optimization problem. Using conditions for optimality, it was reduced to solving a set of nine simultaneous nonlinear equations that we termed the planar motion field equations. Two iterative schemes for solving these equations were given. It was shown that all information in the image concerning motion recovery can be captured by the moments of the image brightness derivatives that constitute the coefficients of the planar motion field equations. These moments are computed during an initial pass over the relevant image regions so that there is no need to refer back to the image after every iteration. This reduces the computation to accumulating 81 moments and performing less than 300 multiplications per iteration in the first iterative scheme and approximately 700 multiplications in the second one.

We also gave a compact proof that the problem can have at most two planar solutions. Through a selected example with synthetic data, it was shown that both schemes may converge to either of the two solutions, depending on the initial condition. In practice, once a solution is obtained, the other can be computed using the equations given for the dual solution.

In the tests carried out, both algorithms have converged to a possible solution, and accurate results have been obtained in less than 40 iterations using the first scheme, and in less than 15 iterations in the second one. As mentioned earlier, the results have not been as satisfactory when the translational motion component is perpendicular to the planar surface. These cases required several hundred iterations of either scheme for accurate solutions. It is conceivable that this special case that results in a unique planar solution can be handled more appropriately by exploiting the fact that the translational motion is in the direction perpendicular to the surface.

Even though both schemes require approximately the same number of computations for convergence to a solution (second scheme converges faster but requires more computation), the second one seems more appropriate for parallel implementation.

We also presented a closed-form solution to the same problem. We first employed the brightness change constraint equation that we developed for planar surfaces to compute 9 intermediate parameters, the elements of a 3×3 matrix, from brightness derivatives at a minimum of eight image points. We referred to them as essential parameters. The special structure of this matrix allows us to compute the motion and plane parameters easily.

References

[1] J. Barron, "A survey of approaches for determining optical flow, environmental layout and egomotion," Univ. Toronto, Toronto, Ont., Canada, Rep. RBCV-TR-84-5, Nov. 1984.

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

[3] C. J. Hay, "Optical motion and space perception, an extention of Gibson's analysis," *Psychol. Rev.*, vol. 73, pp. 550-565, 1966.

[4] E. C. Hildreth, *The Measurement of Visual Motion*. Cambridge, MA: M.I.T. Press, 1983.

[5] B. K. P. Horn and B. G. Schunk, "Determining optical flow," *Artificial Intell.*, vol. 17, pp. 185-203, 1981.

[6] H. C. Longuet-Higgins, "The visual ambiguity of a moving plane," *Proc. Roy. Soc. London*, vol. B 223, pp. 165-175, 1984.

[7] S. J. Maybank, "The angular velocity associated with the optical flow field due to a single moving rigid plane," in *Proc. Sixth European Conf. Artificial Intell.*, Sept. 1984, pp. 641-644.

[8] S. Negahdaripour and B. K. P. Horn, "Direct passive navigation," M.I.T. A.I. Lab., Cambridge, MA, AI Memo 821, Feb. 1985.

[9] R. Y. Tsai, T. S. Huang, and W. L. Zhu, "Estimating three-dimensional motion parameters of a rigid planar patch, II: Singular value decomposition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-30, Aug. 1982.

[10] 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, Jan. 1984.