Direct Passive Navigation

SHAHRAR NEGARHARPOOR 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 formulation. 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 those 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 $3 \times 3$ matrix. The eigenvalue decomposition of the symmetric part of this 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, the optical flow, that is the apparent velocity of image brightness patterns, is used. 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.

In much of the work on recovering surface structure and motion, 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 correspondence, 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 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 $3 \times 3$ 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 $R$ be a point in this plane. Its projection in the image is $r$, where

$$ r = \frac{1}{z} R. $$

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

B. Motion Field

The motion field is the vector field indexed in the image plane by the relative motion of the observer with respect to the environment.
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 $r$ of a point $R$ is given by

$$\frac{dr}{dt} = \frac{1}{R} \cdot \frac{\partial R}{\partial t}. \tag{1}$$

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

$$r = \frac{1}{R} \cdot \frac{\partial R}{\partial t},$$

which can also be written in the compact form

$$r = \left( \frac{1}{R} \right) \cdot \frac{\partial x}{\partial t}, \tag{2}$$

since $a \times (b \times c) = (a \cdot b)c - (a \cdot c)b.$ The vector $r$ lies in the image plane, and so $(r, -t) = 0.$ Further, $r = 0,$ if $[R]R,$ as expected.

Finally, noting that $R = \langle R \cdot E \rangle,$ we get

$$r = \left( \frac{1}{R} \right) \cdot \frac{\partial x}{\partial t}. \tag{3}$$

C. Rigid Body Motion

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

$$R = -\omega \times R - t. \tag{4}$$

Since $R = \langle R \cdot E \rangle,$ we can write this as

$$R = \langle -\omega \cdot E \rangle \times R - t. \tag{5}$$

Substituting for $R,$ in the formula derived above for $r,$ we obtain

$$r = \left( \frac{1}{R} \right) \cdot \frac{\partial x}{\partial t} = \left( \frac{1}{R} \right) \cdot \left( \frac{\partial (x \times R)}{\partial t} - \frac{1}{R} \frac{\partial R}{\partial t} \right). \tag{6}$$

It is important to remember that there is an inherent ambiguity here, since the 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 $R$ and $t$ 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 approximately constant as the surface moves in the environment. If we assume that the image brightness of a patch remains constant, we have

$$\frac{dE}{dt} = 0, \tag{7}$$

or

$$\frac{dE}{dr} \frac{dr}{dt} + \frac{dE}{dt} = 0, \tag{8}$$

where $dE/dr = (dE/dx, dE/dy, 0)^T$ 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

$$E_{r} \cdot r + E_{t} = 0. \tag{9}$$

Substituting for $r,$ we get

$$E_{r} - E_{t} \cdot \left( \frac{1}{R} \right) \left( x \times \omega - \frac{1}{R} \frac{\partial R}{\partial t} \right) = 0. \tag{10}$$

D. Iterative Method

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

$$J = \sum_{I} \left[ c + \omega \cdot R + (r \cdot n)(s \cdot \xi) \right] dx dy$$

where $n/m$ is a unit normal to the plane, and $1/|n|$ is the perpendicular distance of the plane from the origin. Since $R = \langle R \cdot E \rangle,$ we can write this as

$$r - \omega \times R = \langle \omega \cdot E \rangle,$$

so the constraint equation becomes

$$c + \omega \cdot R + (r \cdot n)(s \cdot \xi) = 0. \tag{11}$$

This is the brightness change equation for a plane 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 proportion.

III. RECOVERING MOTION AND STRUCTURE

Given image brightness $E(x, y, t),$ and its spatial and time derivatives, $E_x$ and $E_{xx,t}$ over some region $I$ in the image plane, we are to recover the translational and rotational motions, $t$ and $\omega,$ as well as the plane $n.$ 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 $3 \times 3$ 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

$$J = \sum_{I} \left[ c + \omega \cdot R + (r \cdot n)(s \cdot \xi) \right] dx dy$$

where $n/m$ is a unit normal to the plane, and $1/|n|$ is the perpendicular distance of the plane from the origin. Since $R = \langle R \cdot E \rangle,$ we can write this as

$$r - \omega \times R = \langle \omega \cdot E \rangle,$$

so the constraint equation becomes

$$c + \omega \cdot R + (r \cdot n)(s \cdot \xi) = 0. \tag{11}$$

This is the brightness change equation for a plane 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 proportion.
by suitable choice of the translational and rotational motion vectors \( t \) and \( w \), as well as the normal to the plane \( n \).

For an extremum of \( J \) we must have

\[
\frac{\partial J}{\partial t} = 0, \quad \frac{\partial J}{\partial w} = 0, \quad \text{and} \quad \frac{\partial J}{\partial n} = 0.
\]

That is,

\[
\int_{x} (o \cdot w + (r \cdot n)(o \cdot t) dx dy) = 0, \quad \int_{x} (o \cdot w + (r \cdot n)(o \cdot t) dx dy) = 0, \quad \int_{x} (o \cdot w + (r \cdot n)(o \cdot t) dx dy) = 0.
\]

These equations comprise nine nonlinear (scalar) algebraic equations in terms of the observer motion, \( t \) and \( w \), and the surface normal \( n \). We will call them the planar motion field equations. Some observations about these equations are in order. The first equation is linear in \( w \), \( t \), and \( n \). The second equation is linear in \( w \) and \( t \), but quadratic in \( n \). Finally, the last equation is linear in \( w \) and \( n \), but quadratic in \( t \). 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

\[
\int_{x} (o \cdot t) dx dy = \int_{x} (o \cdot n) (o \cdot t) dx dy = 0,
\]

where

\[
M_{1} = \int_{x} (o \cdot t) dx dy, \quad M_{2} = \int_{x} (o \cdot n) (o \cdot t) dx dy,
\]

\[
d_{1} = \int_{x} c \cdot t dx dy, \quad c_{1} = \int_{x} c \cdot t dx dy.
\]

This can be solved for \( t \) and \( w \), given the surface normal \( n \). The last equation is

\[
N_{2} t = -g,
\]

where

\[
N_{2} = \int_{x} (o \cdot t) dx dy, \quad g = \int_{x} c \cdot (o \cdot w) dx dy.
\]

and can be solved for the surface normal \( n \), given the pair of vectors \( t \) and \( w \).

The motion vectors are given by

\[
w = (M_{1} - M_{2} M_{3}^{T})^{-1} (M_{4} - M_{2} d_{1} - M_{3} d_{1}),
\]

\[
t = -M_{2}^{T} d_{2} + M_{3} d_{2},
\]

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

\[
w = (M_{1} - M_{2} M_{3}^{T})^{-1} (M_{4} - M_{2} d_{1} - d_{2}),
\]

\[
t = -M_{2}^{T} d_{2} + M_{3} d_{2},
\]

The surface normal is simply given by

\[
n = -N_{2} t.
\]

All arrays are either \( 3 \times 3 \) matrices or vectors of length 3, and therefore, the solutions for \( t \), \( w \), and \( n \) 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 \( n \). Using the above equations, we solve for \( t \) and \( w \) in terms of the current value of \( n \), and then for \( n \) in terms of the current values of \( t \) and \( w \). 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 \( t \) and \( w \).

\[
M_{1} = \int_{x} (o \cdot t) dx dy, \quad M_{2} = \int_{x} (o \cdot n) (o \cdot t) dx dy,
\]

\[
M_{3} = \int_{x} (o \cdot n) (o \cdot t) dx dy, \quad M_{4} = \int_{x} (o \cdot w) dx dy.
\]

These may be solved for \( t \) and \( w \). For simplicity, let \( M_{1} \), \( M_{2} \), \( N_{1} \), \( d_{1} \), and \( d_{2} \) be as defined earlier, and let

\[
N_{1} = M_{1}, \quad d_{1} = c_{1},
\]

\[
N_{2} = \int_{x} (o \cdot t) dx dy, \quad d_{2} = \int_{x} c \cdot t dx dy.
\]
Then
\[
(M_1 M_2)^{\alpha} (M_1 M_2)^{\beta} = -d_1 (M_1 M_2)^{\alpha} d_2
\]
and
\[
(N_1 N_2)^{\alpha} (N_1 N_2)^{\beta} = -d_1 (N_1 N_2)^{\alpha} d_2.
\]

The solution of the above equations is given by
\[
u = (M_1 M_2)^{\alpha} (M_1 M_2)^{\beta} (M_1 M_2) - M_1 M_2),
\]
\[
\tau = (M_1 M_2)^{\alpha} (M_1 M_2)^{\beta} (M_1 M_2) - M_1 M_2),
\]
and
\[
\sigma = (N_1 N_2)^{\alpha} (N_1 N_2)^{\beta} (N_1 N_2)^{\alpha} (N_1 N_2)^{\beta} - N_1 N_2).
\]

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 \(\sigma\). We solve for \(\sigma\) and \(\omega\) in terms of the current value of \(\sigma\), and update \(\tau\), then solve for \(\tau\) and \(\omega\) in terms of the current value of \(\tau\), and update \(\sigma\), and, finally, evaluate the improvement in the solution to either continue with the next iteration or 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). 

\[
(M_1)_{\alpha} = \int \psi_{\alpha}(s) x \, ds,
\]
\[
(M_2)_{\beta} = \int \omega_{\alpha}(s) x \, ds,
\]
\[
(N_1)_{\beta} = \int \omega_{\alpha}(s) y \, ds,
\]
\[
(N_2)_{\alpha} = \int \psi_{\alpha}(s) y \, ds,
\]
\[
(d_1)_{\alpha} = \int \psi_{\alpha}(s) z \, ds,
\]
\[
(d_2)_{\beta} = \int \omega_{\alpha}(s) z \, ds,
\]
\[
(N_1)_{\alpha} = \int \omega_{\alpha}(s) z \, ds,
\]
\[
(N_2)_{\beta} = \int \psi_{\alpha}(s) z \, ds,
\]
\[
(e_1) = \int \psi_{\alpha}(s) w \, ds,
\]
\[
(e_2) = \int \omega_{\alpha}(s) w \, ds,
\]
and
\[
(g) = \int \psi_{\alpha}(s) w \, ds + \int \omega_{\alpha}(s) w \, ds.
\]

\(M_1 = N_2\) and \(d_1 = \omega\), do not depend on \(\omega\), \(\tau\), or \(\sigma\), and so need only be computed once. Also, \((e_1, e_2,\bar{e}_1,\bar{e}_2, (\bar{e}_1,\bar{e}_2))\), and \((1, \bar{e}_1,\bar{e}_2))\) depend only on \(r, E_x, E_y\), and \(E_z\), and can be integrated over the image once. This appears to be a set of 5 + 9 + 9 + 27 + 81 = 129 numbers, but, because of symmetry in \((\psi_{\alpha},\omega_{\alpha})\), only 81 numbers have to be stored. These accumulated totals represent all the image information needed to solve the 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 + 162 multiplications to compute \(M_1, d_1, M_2, N_1, N_2\), and \(g\) (note that \(M_1\) and \(N_2\) are symmetric). The updating of \(u, \tau, t, \sigma,\) in comparison, requires 117 multiplications.

In the second scheme, 696 multiplications are carried out at each iteration; we compute the matrices \(M_1, M_2,\) and the vector \(d_1\), 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 \(N_1, N_2\), and the vector \(g\) required in the second half. Further, solving for \(\omega\) and \(\tau\) takes about 270 multiplications.

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 corresponds to an arbitrary uniform surface moving in an arbitrary way. So the brightness gradient, or lack of brightness gradient, \(\psi\), 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 \(\tau\) and \(\omega\) and the planar surface \(n\), is there another motion \(\tau'\) and \(\omega'\) 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 doubtfully 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
\[
c + c' + \omega + (\tau \cdot n)(s \cdot r) = 0,
\]
\[
c + c' + \omega + (\tau' \cdot n')(s \cdot r) = 0.
\]

Subtracting these equations, we get
\[
\omega \cdot (\omega - \omega') + (\tau \cdot n)(s \cdot r) - (\tau' \cdot n')(s \cdot r) = 0.
\]

Now \(\omega = (s \times r, -s \cdot r)\), so
\[
-(s \times r, -s \cdot r) \cdot (\omega - \omega') + (\tau \cdot n)(s \cdot r) - (\tau' \cdot n')(s \cdot r) = 0,
\]
and
\[
-(s \times r, -s \cdot r) \cdot (\omega - \omega') = 0.
\]

If we let \(\omega = (\omega_1, \omega_2, \omega_3)\), then we can write
\[
\omega \times s = \Omega \omega,
\]
where \(\Omega = r_1 s_2 - r_2 s_1, r_1 r_3 - r_3 r_1, -r_2 r_3 + r_3 r_2\) is a suitable \((3 \times 3)\) skew-symmetric matrix. The \(i,j\)-th element of \(\Omega\) equals \(w_{ij}\), where \(w_{ij}\) is the permutation symbol. (It equals +1 when the ordered set \(i,j,k\) is obtained by an even permutation...
are equal.) by an odd permutation, and it is zero if two or more of the indexes where the zero on the right-hand side here represents a 3 matrix of zeros. Now RT taking the transpose of the equation we get

This is to be true for all points r in the image region I and all

products are equal and the above equation is satisfied. and t' is much more constraining. equation involving the dyadic products of n and t as well as of n'. Adding the two equations allows us to eliminate (R

we see immediately that (n

change the lengths of the vectors n and t as long as the product of their lengths remains constant.

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 (R

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, \( n \perp f \). In this case the solution is unique. Further, if \( f \perp 0 \), then \( n \perp 0 \). 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 avoiderrors 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 1 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. 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 70 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 one 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

<table>
<thead>
<tr>
<th>Iter.</th>
<th>(Rotational Par's)</th>
<th>(Translational Par's)</th>
<th>(Surface Par's)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
<td>w1</td>
<td>v1</td>
<td>t1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
<tr>
<td>10</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
<tr>
<td>15</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Iter.</th>
<th>(Rotational Par's)</th>
<th>(Translational Par's)</th>
<th>(Surface Par's)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
<td>w1</td>
<td>v1</td>
<td>t1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
<tr>
<td>10</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
<tr>
<td>15</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Iter.</th>
<th>(Rotational Par's)</th>
<th>(Translational Par's)</th>
<th>(Surface Par's)</th>
</tr>
</thead>
<tbody>
<tr>
<td>No.</td>
<td>w1</td>
<td>v1</td>
<td>t1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
<tr>
<td>10</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
<tr>
<td>15</td>
<td>0.0254</td>
<td>0.0081</td>
<td>0.0051</td>
</tr>
</tbody>
</table>

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

\[ c + (r \times x) \cdot s + \theta = 0 \]

Using the identity \( r \times x = r \times (r \times x) \), we obtain

\[ c - r \cdot (r \times x) + \theta = 0 \]

We now use the isomorphism between vectors and skew-symmetric matrices. Let us define

\[ \Omega = \begin{pmatrix} 0 & -w_2 & w_1 \\ w_2 & 0 & -w_1 \\ -w_1 & w_2 & 0 \end{pmatrix} \]

then, \( \Omega x = (0 \times x) \), and we conclude that

\[ c + r' \Omega x + (\theta + r' \Omega) = 0 \]

or

\[ c + r'(-\Omega + \theta x) = 0 \]

If we define

\[ P = \begin{pmatrix} p_1 & p_2 & p_3 \end{pmatrix} \]

we can finally write

\[ v + r'P = 0 \]

We will refer to \( (P, \theta) \) 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 argument of constraint equation 1 is linear in the elements of \( P \). Several such equations, for different image points, can be used to solve for these parameters. We will show how the special structure of \( P \) can be exploited to recover the motion and plane parameters very easily.

Note that the essential parameters \( P \) are not independent. This is because \( P \) is not an arbitrary \( 3 \times 3 \) 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 \( P \) (and hence \( \Omega \)) three to specify \( \theta \), and another three for \( r \). The matrix \( P \), however, is unchanged if we replace \( P \) by \( kP \) and \( \Omega \) by \( \frac{1}{k} \Omega \) for any nonzero \( k \). Thus, there are actually only eight degrees of freedom, not nine.

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

\[ \det (P + \theta \Omega) = 0 \]

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

\[ p_1(p_2p_3 - p_3p_2) + p_2(p_3p_1 - p_1p_3) + p_3(p_1p_2 - p_2p_1) = 0 \]

We can use this equation, for example, to solve for \( p_3 \) given \( p_1 \) and \( p_2 \). It is difficult to use this equation directly when one attempts to find \( P \) from image brightness measurements.

In the final analysis, the essential parameters are the least number of parameters required to specify the motion and plane parameters, if we are interested in solving for the true motion and plane parameters, that is, the parameters of the line that the camera moved along.
There is a simple way around this problem, however. Note that \( r^T R = 0 \), because \( r = (r_1, \ldots, 0) \times r \). So \( r^T R = 0 \), and
\[
c + r^T R^T = 0,
\]
for arbitrary \( c \). If we let \( P' = P' + I \), we can write
\[
c + r^T R^T = 0,
\]
and conclude that we cannot recover \( P' \) from image brightness measurements alone. To find \( P' \), we must impose the constraint \( det(P + P') = 0 \). 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 \( I \) such that \( P' = P - II \) satisfies
\[
det(P + P') = 0.
\]
Now,
\[
det(P + P') = det(P + P' - 2I) = 0,
\]
so that \( 2I \) must be an eigenvalue of the real symmetric matrix
\[
P^* = P + I.
\]
It will become apparent, in the next section, that we ought to choose \( P^* \) so that
\[
\text{the middle one of the three real eigenvalues of } P^* \text{ for } 2I.
\]
In summary, the overall plan is to find any matrix \( P' \) that satisfies the image brightness constraint equation.

\[
c + r^T R^T = 0,
\]
at a suitable number of image points and consequently determine \( P^* \). We can then solve for the middle eigenvalue of \( P^* \) (which is \( 2I \)) so as to construct the singular matrix \( P' = P - II \), and from that we finally determine \( P' \) and \( I \) as well as \( \Omega \) (and hence \( A \)) using the relationship
\[
P = \Omega + 2 I.
\]

1) Recovering Essential Parameters: We are looking for a matrix \( P' \) that satisfies the brightness-change equation,
\[
c + r^T R^T = 0,
\]
at a chosen number of image points. Now,
\[
r^T R^T = \text{Trace}(\{\sigma a\}^T),
\]
or
\[
r^T R^T = \text{Flat}(\{\sigma a\}) - \text{Flat}(P'),
\]
where \( \text{Flat}(M) \) 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
\[
P' = (p_1, p_2, \ldots, p_m)^T,
\]
\[
a = (r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8)^T.
\]
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 \( a \), such that
\[
a^T P' = -c_i.
\]
As discussed above, there are really only eight independent degrees of freedom. So we can arbitrarily fix one of the components of the vector \( a \). This means that we can solve for the other eight using constraint equations derived from eight image points.

Let \( P' = (p_1, p_2, \ldots, p_m)^T \) denote the solution obtained by setting the last element equal to zero. If we define
\[
P = (p_1, p_2, \ldots, p_m, \delta)^T,
\]
\[
\delta = (r_{12}, r_{13}, r_{14}, r_{15}, r_{16}, r_{17}, r_{18})^T,
\]
then the above constraint equation reduces to
\[
a^T P' = -c_i.
\]
Using eight independent points, we can solve the following linear matrix equation:
\[
\text{A} P' = -c_i,
\]
where
\[
\text{A} = (a_1, a_2, \ldots, a_8)^T, \quad c_i = (c_1, c_2, \ldots, c_8)^T.
\]
The solution of the above equation is
\[
P' = (\text{A}^T \text{A})^{-1} \text{A}^T c_i.
\]
In either case, we construct \( A \) by adjoining a zero to the vector \( P' \). 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 \( I \), \( m \), and \( a \).

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 \( v \) are eigenvectors, and their eigenvalues are \( \mu_v \) and \( \mu_u \), respectively, then
\[
\text{A} u = \mu_u u, \quad \text{A} v = \mu_v v.
\]

Since \( P^* = P - 2I \), then \( u \) and \( (\lambda - 2) v \) are eigenvectors-eigenvalue pairs of \( P^* \), since
\[
P^* u = (P^* - 2I) u = (\lambda - 2) u.
\]

2) Recovering Motion and Structure: We now show how to use the parameters of the translational motion and the plane orientation from the essential parameters. When we have done this, we will be able to find the rotational parameters using
\[
\Omega = u a^T - P.
\]
As we saw before
\[
P^* = P + P' = \text{tn} + a l^T,
\]
since \( \Omega \) is skew-symmetric. Let us use the notation \( u = [a]^T \), and
\[
\gamma = a \cdot l, \quad l = \text{Flat}(l),
\]
and the unit vectors in the directions of the surface normal and the translation vector, respectively. Then,
\[
\text{Trace}(P^*) = \text{Trace}(P') + \text{Trace}(P') = 2 a \cdot l = 2 \gamma.
\]
It turns out that \( \mathbf{u} \) and \( \mathbf{t} \) can be easily recovered from the eigenvectors of the matrix \( \mathbf{P}^* \). In the following lemma, we show that the eigenvectors of \( \mathbf{P}^* \) are combinations of the sought after vectors \( \mathbf{u} \) and \( \mathbf{t} \).

\[ \mathbf{P}^* = \mathbf{U} \mathbf{D} \mathbf{U}^T \]

where \( \mathbf{U} \) is the eigenvector decomposition of the matrix \( \mathbf{P} \). If \( \mathbf{n} \) is not parallel to \( \mathbf{t} \), then,

\[ \mathbf{A} = \text{Diag}( \mathbf{a}(t-1), 0, \mathbf{a}(t+1)) \]

and,

\[ \mathbf{U} = \begin{bmatrix} \mathbf{i} & \mathbf{t} & \mathbf{i} + \mathbf{t} \end{bmatrix} \]

Proof: Note that

\[ \mathbf{P}^* = \mathbf{a}(\mathbf{i} + \mathbf{t}) \]  

Now \((\mathbf{i} + \mathbf{t})\) is the eigenvector with eigenvalue zero since

\[ \mathbf{P}^*(\mathbf{i} + \mathbf{t}) = \mathbf{a}(\mathbf{i} + \mathbf{t})^2 = \mathbf{a}(\mathbf{i} + \mathbf{t}) \]

Since \( \mathbf{P}^* \) is real symmetric, it has three orthogonal eigenvectors. The other two eigenvectors must, therefore, be in the plane containing \( \mathbf{i} \) and \( \mathbf{t} \). Let \( \mathbf{u} = \mathbf{a} \mathbf{i} + \mathbf{b} \mathbf{t} \) and \( \mathbf{v} \) denote an eigenvector-eigenvalue pair for some \( u \) and \( \beta \) (to be determined). Then,

\[ \mathbf{a}(\mathbf{i} + \mathbf{t})^2 \mathbf{u} = \mathbf{a}(\mathbf{i} + \mathbf{t}) \mathbf{v} \]

that becomes

\[ \mathbf{a}(\mathbf{i} + \mathbf{t}) + \mathbf{b}(\mathbf{t} + \mathbf{i}) \mathbf{u} = \mathbf{a}(\mathbf{i} + \mathbf{t}) + \mathbf{b}(\mathbf{t} + \mathbf{i}) \mathbf{v} \]

Since \((\mathbf{i} + \mathbf{t}) = \mathbf{r} \), we can write

\[ \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \]

For this pair of homogeneous equations to have a nontrivial solution for \( u \) and \( \beta \), the determinant of the 2 x 2 coefficient matrix must be zero, that is,

\[ (\mathbf{i} - \mathbf{j}) \mathbf{u} = 0 \]

or

\[ \lambda = \mathbf{i} + \mathbf{j} \pm 1 \]

Substituting for \( \lambda \) into the earlier equations, we obtain

\[ a = \pm \beta \]

Note that \( \mathbf{e} = \mathbf{i} \mathbf{e} \mathbf{n} < 0 \) and \( \mathbf{e} = \mathbf{i} \mathbf{e} \mathbf{n} + 1 > 0 \) because \( t < 1 \), as it is the cosine of the angle between \( \mathbf{n} \) and \( \mathbf{t} \). So one eigenvalue is neg-

\[ \mathbf{P}^* = \mathbf{a}(\mathbf{i} + \mathbf{t}) \mathbf{b}(\mathbf{i} + \mathbf{t}) \mathbf{c}(\mathbf{i} + \mathbf{t}) \]

Since \((\mathbf{i} + \mathbf{t}) = \mathbf{r} \), we can write

\[ \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \]

Note that we can determine \( \mathbf{n} = \mathbf{a}[\mathbf{e}] \) from

\[ \mathbf{a} = \frac{1}{\mathbf{i} - \mathbf{j}} \lambda - \mathbf{i} \lambda \]

The equations for \( u_1 \) and \( u_2 \) are linear in \( \mathbf{i} \) and \( \mathbf{t} \), and so can be easily solved for these vectors:

\[ \mathbf{A} = \sqrt{\mathbf{i}^2 + \mathbf{t}^2} \mathbf{a} + \sqrt{\mathbf{i}^2 - \mathbf{t}^2} \mathbf{b} \]

\[ \mathbf{B} = \sqrt{\mathbf{i}^2 + \mathbf{t}^2} \mathbf{b} + \sqrt{\mathbf{i}^2 - \mathbf{t}^2} \mathbf{a} \]

The signs of the eigenvectors are arbitrary. If we change the sign of \( u_1 \), we obtain instead

\[ \mathbf{A} = \sqrt{\mathbf{i}^2 + \mathbf{t}^2} \mathbf{a} + \sqrt{\mathbf{i}^2 - \mathbf{t}^2} \mathbf{b} \]

\[ \mathbf{B} = \sqrt{\mathbf{i}^2 + \mathbf{t}^2} \mathbf{b} + \sqrt{\mathbf{i}^2 - \mathbf{t}^2} \mathbf{a} \]

where \( \mathbf{A} \) and \( \mathbf{B} \) 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 \( \mathbf{A} \) and \( \mathbf{B} \). We show next that two of these solutions can be dis-

\[ \mathbf{A}^* = \mathbf{a}(\mathbf{i} + \mathbf{t}) \]

\[ \mathbf{B}^* = \mathbf{a}(\mathbf{i} + \mathbf{t}) \]

Now \((\mathbf{i} + \mathbf{t})\) is the eigenvector with eigenvalue zero since

\[ \mathbf{P}^*(\mathbf{i} + \mathbf{t}) = \mathbf{a}(\mathbf{i} + \mathbf{t})^2 = \mathbf{a}(\mathbf{i} + \mathbf{t}) \]

Since \( \mathbf{P}^* \) is real symmetric, it has three orthogonal eigenvectors. The other two eigenvectors must, therefore, be in the plane containing \( \mathbf{i} \) and \( \mathbf{t} \). Let \( \mathbf{u} = \mathbf{a} \mathbf{i} + \mathbf{b} \mathbf{t} \) and \( \mathbf{v} \) denote an eigenvector-eigenvalue pair for some \( u \) and \( \beta \) (to be determined). Then,

\[ \mathbf{a}(\mathbf{i} + \mathbf{t})^2 \mathbf{u} = \mathbf{a}(\mathbf{i} + \mathbf{t}) \mathbf{v} \]

that becomes

\[ \mathbf{a}(\mathbf{i} + \mathbf{t}) + \mathbf{b}(\mathbf{t} + \mathbf{i}) \mathbf{u} = \mathbf{a}(\mathbf{i} + \mathbf{t}) + \mathbf{b}(\mathbf{t} + \mathbf{i}) \mathbf{v} \]

Since \((\mathbf{i} + \mathbf{t}) = \mathbf{r} \), we can write

\[ \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \]

For this pair of homogeneous equations to have a nontrivial solution for \( u \) and \( \beta \), the determinant of the 2 x 2 coefficient matrix must be zero, that is,

\[ (\mathbf{i} - \mathbf{j}) \mathbf{u} = 0 \]

or

\[ \lambda = \mathbf{i} + \mathbf{j} \pm 1 \]

Substituting for \( \lambda \) into the earlier equations, we obtain

\[ a = \pm \beta \]

Note that \( \mathbf{e} = \mathbf{i} \mathbf{e} \mathbf{n} < 0 \) and \( \mathbf{e} = \mathbf{i} \mathbf{e} \mathbf{n} + 1 > 0 \) because \( t < 1 \), as it is the cosine of the angle between \( \mathbf{n} \) and \( \mathbf{t} \). So one eigenvalue is neg-
If this is to be true for all vectors $x$, we must have

$$
\Omega' = \Omega - \mu_2^T + \mu_3^T.
$$

So, we finally obtain

$$
-\Omega' + \mu_2^T = -\Omega + \mu_2^T - \mu_2^T + \mu_3^T,
$$

or,

$$
-\Omega' + \mu_2^T = -\Omega + \mu_2^T = P.
$$

We conclude that $n', t'$, and $o'$, 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 employed the brightness change constraint equation that we developed for planar surfaces to compute 9 intermediate parameters, the elements of a $3 \times 3$ 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


