# COMPUTATIONALLY-EFFICIENT METHODS FOR RECOVERING TRANSLATIONAL MOTION

Berthold K.P. Horn* - E. J. Weldon, Jr.**

Artificial Intelligence Lab
Massachusetts Institute of Technology
545 Technology Square
Cambridge, Massachusetts 02139

Department of Electrical Engineering
University of Hawaii at Manoa
Honolulu, Hawaii 96822

## ABSTRACT

We have developed robust methods for determining the motion of an observer translating through a static environment. These methods also apply to the case of arbitrary observer motion when the rotational component is known. (The case of purely rotational motion is treated elsewhere (1)). Our methods do not rely on establishing point correspondence, nor do they determine optical flow. They employ only first derivatives of the image brightness function and do not assume an analytic form for the imaged surface.

Our methods are all based on minimizing the difference between the observed time derivative of the brightness and that predicted from the observed spatial brightness gradient given an estimated motion. The methods exploit the fact that any imaged surface must lie in front of the camera. They are robust in that all points in the image contribute to the final determination of motion.

## 1. Introduction

In this paper we consider the problem of determining the motion of a monocular observer moving with respect to a rigid, unknown world. We use a "least squares", as opposed to a discrete method of solving for the motion parameters; our method uses all of the points in a two-image sequence and does not attempt to establish correspondence between the images. Hence the method is fairly robust with respect to quantization error, noise, illumination gradients, and other effects.

We can determine the observer motion in two special cases:

a) when the motion is pure translation or when the rotational component of the motion is known.

b) when the motion is pure rotation; this case is discussed in Reference 1.

At this writing we have not developed a robust method which is applicable to arbitrary motion.

Three methods of determining observer motion are considered in this paper:

a. Minimization of the Integral of $Z^2$
Given an estimate of motion, the depth at most world points can be calculated. Using an incorrect value for the motion gives an estimate of depth which is inaccurate and often grossly so. For scenes with a limited depth range, an accurate estimate of motion is obtained by minimizing the sum of the squares of the depth estimates over the image.

b. Minimization of the Number of Outliers
For some points in the image, the estimated scene depth is negative. Such "outliers" may be caused by noise, quantization error, scene occlusion, under-sampling, and errors in the estimates of brightness derivatives. An accurate estimate of motion is obtained by minimizing the number of such points.

c. The Perceptron "Learning" Algorithm In this iterative method, an image point is found for which the calculated depth, based on the current estimate of the motion, is negative. The motion estimate is then adjusted to make this depth value non-negative. Under reasonable conditions, this algorithm converges to the correct motion in a finite number of steps.

We show that the field of view should be large to accurately recover the components of motion in the direction towards the image region. We also demonstrate the importance of points where the time derivative of brightness is small, and discuss difficulties resulting from very large depth ranges. We emphasize the need for adequate filtering of the image data before sampling to avoid aliasing, both in the spatial and temporal dimensions.

The algorithms have been applied to real image sequences. The results show that subject to the limitations mentioned above, this approach offers a computationally efficient method of determining translational motion.

### 1.1 Earlier Work

In the continuous, or least squares, approach to motion vision, motion parameters are found which are consistent with the observed motion of the entire image. Bruss and Horn [2] use this approach to calculate motion parameters assuming that the optical flow is known at each point. Adiv [3] applies the results of Bruss and Horn to a world consisting of independently moving planar objects; he shows that given the optical flow, segmentation can be performed and the motion of the objects can be calculated. Negahdaripour and

Horn [4] eschew the use of optical flow and calculate the observer's motion directly from the spatial and temporal derivatives of the image brightness, assuming a planar world. The advantage of this direct approach, which we also use here, is that certain computational difficulties inherent in the calculation of optical flow are avoided. Specifically, it is not necessary to make the usual assumption that the optical flow field is smooth; an assumption which is violated at object boundaries.

Waxman and Ullman [5] also avoid the discrete approach to motion vision; their technique makes use of various first and second derivatives of the image to compute both the motion parameters and the structure of the imaged world. In the interests of developing methods which can be implemented, the techniques presented in this paper avoid the use of second and higher-order derivatives.

### 1.2 Summary of the Paper

Our approach to the motion vision problem can be summarized as follows: Given the observer motion and the spatial brightness function of the image, one can predict the time derivative of brightness at each point in the image. We find the motion that minimizes the integral of the square of the difference between this predicted value and the observed time derivative. The integral is taken over the image region of interest, which, in the discussion here, is usually taken to be the whole image.

We use auxiliary vectors derived from the derivatives of brightness and the image position that occur in the basic brightness change constraint equation. Study of the distribution of the directions of these vectors on the unit sphere suggests specific algorithms and also helps uncover relationships between accuracy and parameters of the imaging situation.

We have developed several algorithms for recovering the translational velocity in the case of pure translation. These algorithms exploit the constraint that objects have to be in front of the camera in order to be imaged. This constraint leads to a non-linear optimization problem. The performance of these algorithms depends on a number of factors including:

- the angle subtended by the image, i.e., the field of view
- the direction of motion relative to the optical axis
- the depth range of the scene
- the spatial frequency content of the image
- the noise in the estimated time derivative of brightness
- the noise in the estimated spatial derivatives of brightness
- the number of picture cells considered.

We have not yet been able to select a "best" algorithm from the set developed, since one may be most accurate under one set of circumstances while another is best in a different situation. Also, the better algorithms tend to require more computation, and some do not lend themselves to parallel implementation. Further study and experimentation with real image data will be needed to determine the range of applicability of each algorithm.

### 1.3 Comments on Sampling, Filtering and Aliasing

Work with real image data has demonstrated the need to take care in filtering and sampling. The estimates of spatial and time derivatives are sensitive to aliasing effects resulting from inadequate low-pass filtering before sampling. This is easily overlooked, particularly in the time direction. It is usually a mistake, for example, to simply pick every n-th frame out of an image sequence. At the very least, n consecutive frames should be averaged (i.e., added) before sampling in order to reduce the high frequency components. One may object to the "smearing" introduced by this technique, but a series of widely separated snap-shots typically do not obey the conditions of the sampling theorem, and as a result the estimates of the derivatives may contain large errors.

This, of course, is nothing new, since the same considerations apply when one tries to estimate the optical flow using first derivatives of image brightness (Horn and Schunck [6]). It is important to remember that filtering must be done before sampling--once the data has been sampled, the damage has been done.

## 2. The Brightness-Change Constraint Equation

Following Longuet-Higgins and Prazdny [7] and Bruss and Horn [2] we use a viewer-based coordinate system. Figure 1 depicts the system under consideration. A world point

$$\underset{\sim}{R} = (X,Y,Z)^T \tag{1}$$

is imaged at

$$\underset{\sim}{r} = (x,y,1)^T \tag{2}$$

That is, the image plane has equation $Z = 1$. The origin is at the projection center and the Z-axis runs along the optical axis. The X and Y axes are parallel to the x and y axes of the image plane. Image coordinates are measured relative to the principal point, the point $(0,0,1)^T$ where the optical axis pierces the image plane. The points $\underset{\sim}{r}$ and $\underset{\sim}{R}$ are related by the perspective projection equation

$$\underset{\sim}{r} = (x,y,1)^T = \left(\frac{X}{Z}, \frac{Y}{Z}, 1\right)^T = \frac{\underset{\sim}{R}}{\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}}} \tag{3}$$

where

$$Z = \underset{\sim}{R}\cdot\hat{\underset{\sim}{z}} \tag{4}$$

and where $\hat{\underset{\sim}{z}}$ denotes the unit vector in the Z direction.

If the observer moves with instantaneous translational velocity $\underset{\sim}{t} = (U,V,W)^T$ and instantaneous rotational velocity $\underset{\sim}{\omega} = (A,B,C)^T$ relative to a fixed environment, then the time derivative of the vector $\underset{\sim}{R}$ can be written as

$$\underset{\sim}{R}_t = -\underset{\sim}{t} - \underset{\sim}{R}\times\underset{\sim}{\omega} . \tag{5}$$

The motion of the world point R results in motion of the corresponding image point; the value of this motion field is given by

$$\underset{\sim}{r}_t = \frac{d\underset{\sim}{r}}{dt} = \frac{d}{dt}\left(\frac{\underset{\sim}{R}}{\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}}}\right) = \frac{\underset{\sim}{R}_t(\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}}) - (\underset{\sim}{R}_t\cdot\hat{\underset{\sim}{z}})\underset{\sim}{R}}{(\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}})^2} \qquad (6)$$

Then since $\underset{\sim}{A} \times (\underset{\sim}{B} \times \underset{\sim}{C}) = (\underset{\sim}{C}\cdot\underset{\sim}{A})\underset{\sim}{B} - (\underset{\sim}{A}\cdot\underset{\sim}{B})\underset{\sim}{C}$ this can also be expressed as

$$\underset{\sim}{r}_t = \frac{\hat{\underset{\sim}{z}} \times (\underset{\sim}{R}_t \times \underset{\sim}{r})}{(\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}})^2} . \qquad (7)$$

Substituting Eq. (5) into this result gives [4]

$$\underset{\sim}{r}_t = -(\hat{\underset{\sim}{z}} \times (\underset{\sim}{r} \times (\underset{\sim}{r} \times \underset{\sim}{\omega} - \frac{\underset{\sim}{t}}{\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}}}))) \qquad (8)$$

In component form this can be expressed as

$$\underset{\sim}{r}_t = \begin{bmatrix} x_t \\ y_t \\ 0 \end{bmatrix} = \begin{pmatrix} \frac{-U+xW}{Z} + Axy - B(x^2+1) + Cy \\ \frac{-V+yW}{Z} - Bxy + A(y^2+1) - Cx \\ 0 \end{pmatrix} \qquad (9)$$

a result first obtained by Longuet-Higgins and Prazdny [7].

This shows that given the world motion, the motion field can be calculated for every image point. If we assume that the brightness of a small surface patch is not changed by motion, then expansion of the total derivative of brightness E leads to

$$\frac{\partial E}{\partial x} x_t + \frac{\partial E}{\partial y} y_t + \frac{\partial E}{\partial t} = 0 \qquad (10)$$

Now denoting the vector $(\partial E/\partial x,\ \partial E/\partial y,\ 0)$ by $E_r$ and $\partial E/\partial t$ by $E_t$ permits us to express this result more compactly in the form

$$E_r \cdot \underset{\sim}{r}_t + E_t = 0 \qquad (11)$$

Substituting Eq. (8) into this result and rearranging gives

$$E_t - (((E_r \times \hat{\underset{\sim}{z}}) \times \underset{\sim}{r}) \times \underset{\sim}{r}) \cdot \underset{\sim}{\omega} + \frac{((E_r \times \hat{\underset{\sim}{z}}) \times \underset{\sim}{r}).\underset{\sim}{t}}{\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}}} \qquad (12)$$

Now to simplify this expression we let

$$\underset{\sim}{s} = ((E_r \times \hat{\underset{\sim}{z}}) \times \underset{\sim}{r}) \qquad (13)$$

and

$$\underset{\sim}{v} = -\underset{\sim}{s} \times \underset{\sim}{r} \qquad (14)$$

so Eq. (12) reduces to the "Brightness Change Constraint Equation" of Negahdaripour and Horn [4]

$$\underset{\sim}{v} \cdot \underset{\sim}{\omega} + \frac{\underset{\sim}{s}.\underset{\sim}{t}}{\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}}} = -E_t \qquad (15)$$

The vectors $\underset{\sim}{s}$ and $\underset{\sim}{v}$ can be expressed in component form as

$$\underset{\sim}{s} = \begin{bmatrix} -E_x \\ -E_y \\ xE_x + yE_y \end{bmatrix} \text{ and } \underset{\sim}{v} = \begin{bmatrix} E_y + y(xE_x + yE_y) \\ -E_x - x(xE_x + yE_y) \\ yE_x - xE_y \end{bmatrix} \qquad (16)$$

Note that $\underset{\sim}{s} \cdot \underset{\sim}{r} = 0$ and $\underset{\sim}{v} \cdot \underset{\sim}{r} = 0$ and $\underset{\sim}{s} \cdot \underset{\sim}{v} = 0$. These three vectors thus form an orthogonal triad. The vectors $\underset{\sim}{s}$ and $\underset{\sim}{v}$ are inherent static properties of the image. Note that the projection of $\underset{\sim}{s}$ in the image plane is just the (negative) gradient of the image. Also, the quantity $\underset{\sim}{s}$ indicates the direction in which translation of a given magnitude will contribute maximally to the temporal brightness change of a given pixel. The quantity $\underset{\sim}{v}$ plays a similar role for rotation.

## 3. Solving the Brightness Change Constraint Equation

Equation (15) relates observer motion $(\underset{\sim}{t},\underset{\sim}{\omega})$, the depth of the world $\underset{\sim}{R}\cdot\hat{\underset{\sim}{z}} = Z(x,y)$, and certain measurable quantities of the image $(\underset{\sim}{s},\underset{\sim}{v})$. In general, it is not possible to solve for the first two of these given the last. Some interesting special cases are addressed in this paper, in Horn and Weldon [1], and in Negahdaripour and Horn [4].

In this paper we assume that the rotation $\underset{\sim}{\omega}$ is known; then we show that the translation $\underset{\sim}{t}$ can be determined under appropriate conditions. Once $\underset{\sim}{t}$ is known, the brightness change constraint equation can be used to find the depth at each pixel

$$Z = \underset{\sim}{R}.\hat{\underset{\sim}{z}} = -\frac{\underset{\sim}{s}.\underset{\sim}{t}}{E_t + \underset{\sim}{v}.\underset{\sim}{\omega}} \qquad (17)$$

In this paper we consider various integrals and sums over an image region thought to correspond to a single rigid object in motion relative to the viewer. In the simplest case, the observer is moving relative to a static environment and the whole image can be used. The size of the field of view has a strong effect on the accuracy of the determination of the components of motion along the optical axis. When we need to estimate this accuracy we will for convenience assume a circular image of radius $r_v$. This corresponds to a conical field of view with half angle $\theta_v$, where $r_v = \tan\theta_v$, since we have assumed that the focal length equals one. (We assume that $0 < \theta_v < \pi/2$).

We will show that the field of view should be large. Although orthographic projection usually simplifies machine vision problems, this is one case in which we welcome the effects of perspective "distortion"!

If the rotation vector is known, perhaps measured by some other instrument, then the brightness change constraint equation (15) reduces to

$$(1/Z)\ \underset{\sim}{s}\cdot\underset{\sim}{t} = -E_t' \qquad (18)$$

where

$$E_t' = E_t + \underset{\sim}{v}\cdot\underset{\sim}{\omega} \qquad (19)$$

In the remainder of this paper we assume that $\underset{\sim}{\omega}$ is known and do not distinguish between $E_t$ and $E_t'$.

When depth is known, it is straightforward to recover the motion. We cannot, in general, find a motion to satisfy the brightness change constraint equation at every picture cell because of noise in the measurements. Instead we minimize the integral of the error in brightness

$$\iint [E_t + (1/Z)\underset{\sim}{s}\cdot\underset{\sim}{t}]^2 \qquad (20)$$

Differentiating with respect to $\underset{\sim}{t}$ and setting the result equal to 0 leads to the vector equation

$$[[\iint (1/Z)^2 \underset{\sim}{s}\,\underset{\sim}{s}^T]\underset{\sim}{t} = -\iint(1/Z)\, E_t \underset{\sim}{s} \qquad (21)$$

This is a set of three linear equations in the three components of $\underset{\sim}{t}$. The coefficient matrix is symmetric and only the right-hand side depends on the time derivative of brightness. Note that in Eq. (21) we attach less weight to information from points where Z is large.

The method is robust if the correct values of depth are given. If estimates are used, the quality of the result will depend on the quality of the estimates. The accuracy of the result also depends on the size of the field of view, as we show later.

### 3.1 Distribution of the Directions of $\underset{\sim}{s}$

To understand the properties of the above algorithm for recovering $\underset{\sim}{t}$ we must examine the matrix obtained by integrating multiples of $\underset{\sim}{s}\,\underset{\sim}{s}^T$. We can think of the direction of $\underset{\sim}{s}$ as identifying a point on the unit sphere and of a multiple of $||\underset{\sim}{s}||$ as the mass of a particle placed there. The integral considered is related to the inertia matrix of the set of particles on the unit sphere.

We know that $\underset{\sim}{s}\cdot\underset{\sim}{r} = 0$ and that the possible directions of $\underset{\sim}{r}$ lie within the field of view. For a particular value of $\underset{\sim}{r}$, the equation $\underset{\sim}{s}\cdot\underset{\sim}{r} = 0$ defines a plane that cuts the unit sphere in a great circle (see Fig. 2). The vector $\underset{\sim}{s}$ must point in a direction corresponding to a point on this great circle. Since $\underset{\sim}{r}$ lies inside a cone of directions with half-angle $\theta_v$, these great circles have axes that lie in this cone also. The collection of great circles lies in a band around the unit sphere of width equal to the total width of the visual field.

We can obtain the same result algebraically as follows: Let $\hat{\underset{\sim}{r}}$, $\hat{\underset{\sim}{v}}$, and $\hat{\underset{\sim}{s}}$ be unit vectors in the orthogonal directions $\underset{\sim}{r}$, $\underset{\sim}{v}$, and $\underset{\sim}{s}$. Then

$$(\hat{\underset{\sim}{z}}\cdot\hat{\underset{\sim}{r}})^2 + (\hat{\underset{\sim}{z}}\cdot\hat{\underset{\sim}{v}})^2 + (\hat{\underset{\sim}{z}}\cdot\hat{\underset{\sim}{s}})^2 = 1 \qquad (24)$$

while

$$(\hat{\underset{\sim}{x}}\cdot\hat{\underset{\sim}{s}})^2 + (\hat{\underset{\sim}{y}}\cdot\hat{\underset{\sim}{s}})^2 + (\hat{\underset{\sim}{z}}\cdot\hat{\underset{\sim}{s}})^2 = 1 \qquad (25)$$

so

$$(\hat{\underset{\sim}{x}}\cdot\hat{\underset{\sim}{s}})^2 + (\hat{\underset{\sim}{y}}\cdot\hat{\underset{\sim}{s}})^2 = (\hat{\underset{\sim}{z}}\cdot\hat{\underset{\sim}{r}})^2 + (\hat{\underset{\sim}{z}}\cdot\hat{\underset{\sim}{v}})^2 \qquad (26)$$

but

$$(\hat{\underset{\sim}{z}}\cdot\hat{\underset{\sim}{r}})^2 > \cos^2\theta_v \text{ and } (\hat{\underset{\sim}{z}}\cdot\hat{\underset{\sim}{v}})^2 > 0 \qquad (27)$$

so

$$(\hat{\underset{\sim}{x}}\cdot\hat{\underset{\sim}{s}})^2 + (\hat{\underset{\sim}{y}}\cdot\hat{\underset{\sim}{s}})^2 > \cos^2\theta_v \qquad (28)$$

Thus the directions of s lie within an angle $\theta_v$ of the "equator" of the unit sphere. We call this band the permissible band.

Figure 3 shows front and back views of the permissible band of the real 64 by 64 pixel image shown in Figure 5 with $\theta_v = 27^o$. Figure 4 shows the same band with $\theta_v = 15^o$. In each case the signal to noise ratio is 40 db and 8-b+t gray-level quantization is used.

### 3.2 Ensemble Average of the Integral of $\underset{\sim}{s}\,\underset{\sim}{s}^T$

The integral of $\underset{\sim}{s}\,\underset{\sim}{s}^T$ varies from image to image. However, we can obtain a better understanding of this integral by averaging over an ensemble of images with all possible directions for the brightness gradient at each image point. We assume that different directions for the brightness gradient are equally likely. The result so obtained can be viewed in another way: it is the integral obtained in the limit from a textured image as the scale of the texture is made smaller and smaller. In this case we can arrange for every direction of the brightness gradient to be found in any small patch of the image. By suitable choice of the texture we can arrange that no direction of the brightness gradient is favored--all directions occur with equal frequency. If we take into account the distribution of directions of $\underset{\sim}{s}$ and the weights $||\underset{\sim}{s}||$, we find (in Reference 1) that

$$\overline{\iint \underset{\sim}{s}\,\underset{\sim}{s}^T} = k_s \begin{vmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & r_v^2/2 \end{vmatrix} \qquad (29)$$

where the constant $k_s$ depends on the size of the field of view and the distribution of magnitudes of the brightness gradient. In practice we can find $k_s$ by noting that

$$\text{Trace}\ \underset{\sim}{s}\,\underset{\sim}{s}^T = \text{Trace}(\underset{\sim}{s}\,\underset{\sim}{s}^T) = \underset{\sim}{s}\cdot\underset{\sim}{s} \qquad (30)$$

so

$$2k_s(1 + r_v^2/4) = \overline{\iint \underset{\sim}{s}\cdot\underset{\sim}{s}} \qquad (31)$$

Note that the condition number, the ratio of the largest to smallest eigenvalue, is just $\min(r_v^2/2,\ 2/r_v^2)$ which reaches a minimum of 1 when $r_v = \sqrt{2}$. In the case of pure translation then, the component of translation along the optical axis is found with less accuracy than the other two components when the field of view has a half-angle narrower than $\theta_v = \tan^{-1}\sqrt{2} = 54.74...$ degrees.

### 3.3 A Simplified Method of Calculating $\underset{\sim}{t}$ When Depth is Known

While the method of determining $\underset{\sim}{t}$ given in Eq. [21] is simple enough, we may gain additional insight from a simplified version based on the average integral for $\underset{\sim}{s}\,\underset{\sim}{s}^T$ developed in the previous section. Inverting the matrix of Eq. (29) and substituting into the solution for $\underset{\sim}{t}$ given in Eq. (22) yields

$$\begin{aligned} U &\approx (1/k_s)\iint E_x E_t Z^2 \\ V &\approx (1/k_s)\iint E_y E_t Z^2 \\ W &\approx (1/k_s)(2/r_v^2)\iint (xE_x + yE_y)E_t Z^2 \end{aligned} \qquad (32)$$

where

$$2k_s(1+r_v^2/y) = \iint (E_x^2 + E_y^2 + (xE_x + yE_y)^2)Z^2 \qquad (33)$$

The first three integrands are just the three components of $-ZE_t\underset{\sim}{s}$ while the integrand in the expression for $k_s$ is $\left\| Z\,\underset{\sim}{s}\right\|^2$. In like manner we can derive simplified formulae of this kind for the solutions of Eqs. (26) and (28). Experiments with synthetic image data suggest that the estimate of $\underset{\sim}{t}$ given by the above approximation is considerably less accurate than that obtained by actually computing the integral of $\underset{\sim}{s}\,\underset{\sim}{s}^T$. It is nevertheless of interest to note the dependence of U on the integral of a multiple of the product $E_xE_t$ and the dependence of V on the integral of a multiple of the product $E_yE_t$. Further, one may

note that the integral of a multiple of $(xE_x + yE_y)$ underestimates W and so has to be magnified by the factor $2/r_v^2$, along with any small errors it may contain.

## 4.0 Solving the Brightness Change Constraint Equation with Rotation Known

In this section we deal with the problem of determining the direction of translation given the rotation vector $\underset{\sim}{\omega}$.

### 4.1 The Importance of a Wide Field of View

In the general case, the need for a wide field of view is very clear. In a small image region near the center of the image, for example, rotation about the y-axis looks the same as translation along the x-axis, while rotation about the x-axis looks the same as translation along the (negative) y-axis. As is well known in photogrammetry, a large field of view is needed to separate these components of motion [11,12].

If we take note of this ambiguity, and the uncertainty with which the components of rotation and translation along the optical axis can be determined, we see that locally, out of six parameters, only two combinations can be estimated. These two quantities are just the components of the motion field. The argument can also be made for points at some distance from the principal point of the image.

There is a difference between the case when the motion is predominantly along the optical axis and the case where it is predominantly parallel to the image plane. The transition between the two situations occurs when the direction of the vector $\underset{\sim}{t}$ moves outside the cone of directions of the field of view, that is, when the focus of expansion (or compression) moves outside the image. When the focus of expansion (FOE) is inside the image, then the great circle defined by $\underset{\sim}{s}\cdot\underset{\sim}{t} = 0$ lies entirely inside the permissible band on the unit sphere. The measured values of $\bar{\underset{\sim}{s}}$ then provide constraint all the way around the great circle. Conversely, when the focus of expansion is outside the image, then the great circle cuts the permissible band at an angle greater than $\theta_v$. In this case the known values of $\underset{\sim}{s}$ provide constraint only along two sections of the great circle. These sections get shorter and shorter as the vector $\underset{\sim}{t}$ becomes more and more parallel to the image plane. It should be clear that in this case the direction of the vector $\underset{\sim}{t}$ can be determined with less accuracy than when the focus of expansion is near the principal point (especially if $\theta_v$ is small.)

### 4.2 The s-Bar Projection

The integrals on the right hand side of the equations for $\underset{\sim}{t}$ developed in Section 3.1 contain positive multiples of the vector

$$\bar{\underset{\sim}{s}} = -\text{sign}(E_t)\ \underset{\sim}{s} \qquad (34)$$

(Here we only care about the directions of the vectors so we ignore scale factors). Now in the case of translation with known rotation,

$$E_t = -(1/Z)\underset{\sim}{s}\cdot\underset{\sim}{t}$$

and since $Z > 0$,

$$\bar{\underset{\sim}{s}}\cdot\underset{\sim}{t} = (1/Z)\ \text{sign}(\underset{\sim}{s}\cdot\underset{\sim}{t})\ (\underset{\sim}{s}\cdot\underset{\sim}{t}) = (1/Z)\ |\underset{\sim}{s}\cdot\underset{\sim}{t}| \geq 0 \qquad (35)$$

We are only interested at this point in the sign of $\bar{\underset{\sim}{s}}\cdot\underset{\sim}{t}$, so we can use any convenient positive multiple of $\bar{\underset{\sim}{s}}\cdot\underset{\sim}{t}$ such as

$$-(1/E_t)\underset{\sim}{s},\quad -\text{sign}(E_t)\underset{\sim}{s},\quad \text{or}\quad -E_t\underset{\sim}{s}$$

in the discussion that follows.

Equation (35) states that $\bar{\underset{\sim}{s}}$ can only lie in the hemisphere that has $\underset{\sim}{t}$ as its navel. We call this the compatible hemisphere in the case of translation with known rotation. Since $\bar{\underset{\sim}{s}}$ is a multiple of $\underset{\sim}{s}$, it must also lie in the permissible band. Thus $\bar{\underset{\sim}{s}}$ can only lie in the intersection of the permissible band and the compatible hemisphere. We will exploit this geometric insight shortly.

Figures 6 and 7 show examples of $\bar{\underset{\sim}{s}}$ spheres. Figure 6 shows the permissible band on the $\bar{\underset{\sim}{s}}$ sphere for the image of figure 5 with the FOE near the image center. Figure 7 shows the $\bar{\underset{\sim}{s}}$-sphere for the same image with the FOE considerably outside the image ($\theta$=45°).

Our task can be viewed as that of finding the hemisphere that contains all of the directions specified by the vectors $\bar{\underset{\sim}{s}}$ derived from the image. Note that the solution may not be unique and, even worse, there may not be any solution. Later we will modify the problem definition somewhat to deal with these possibilities.

### 4.3 Motion Determination as a Linear Programming Problem

We wish to find a vector $\underset{\sim}{t}$ that makes $\bar{\underset{\sim}{s}}\cdot\underset{\sim}{t} > 0$ at all image points. We can think of this as a gigantic linear programming problem. There are three unknowns and one inequality for every picture cell. Actually, since we do not care about the magnitude of $\underset{\sim}{t}$, there are only two degrees of freedom.

Since we do not have a criterion function to be extremized, we will have an infinite number of solutions--if there are any solutions at all. All of these solutions will lie in a convex polygon on the unit sphere. The sides of this polygon are portions of great circles corresponding to constraints which we will call critical constraints. With data from a large number of cells we expect this solution polygon to be small. We may choose its center as the "best" solution.

Typically the solution polygon will have relatively few sides. Thus data from a small number of critical picture cells fully constrains the solution. First of all, note that each side of this polygon corresponds to an equality of the form $\bar{\underset{\sim}{s}}\cdot\underset{\sim}{t} = 0$ for some picture cell. From the brightness change constraint equation [15], we know that $E_t = 0$ when $\bar{\underset{\sim}{s}}\cdot\underset{\sim}{t} = 0$. Thus the critical constraints are provided by picture cells where $E_t$ is small (and $\|\bar{\underset{\sim}{s}}\|$ is not). This is an important observation which can be used to reduce the size of the linear programming problem; we simply disregard the inequalities arising from picture cells where $|E_t|$ is large.

Unfortunately there is a class of points for which $\bar{s}\cdot\underset{\sim}{t}$ is arbitrary even though $E_t$ is small (and $\|\bar{s}\|$ is not); these are image points for which Z is large. Such points provide "false" constraints on $\underset{\sim}{t}$. For a practical system some means must be found for identifying these points. One way of doing this for images with large depth range is based on the following observation. In a real image, regions for which Z is large (i.e., the background) tend to encompass a significant area with all points in the area having $E_t \neq 0$. On the contrary, points with $\bar{s}\cdot\underset{\sim}{t} = 0$ and $\|\bar{s}\|$ large are usually isolated and surrounded by regions for which $E_t \neq 0$. The above difficulty appears in all of the methods of determining motion discussed in this paper; it is harder to determine $\underset{\sim}{t}$ when the depth range is large.

We observe in passing that the points that are most useful in constraining the translational motion vector are the very same points where it is difficult to calculate depth accurately!

The linear programming method of determining $\underset{\sim}{t}$ discussed above uses relatively little of the image data. In fact, only points at the edge of the compatible hemisphere influence the solution at all. While this is a sensible procedure if the data is perfect, it will be quite sensitive to noise. In fact for real, that is, noisy images, there will normally be no hemisphere which contains all of the points on the $\bar{s}$ sphere, and so the linear programming method will necessarily fail. It seems clear that any practical method of finding $\underset{\sim}{t}$ must use a large number of image points so that it will not be compromised by a few inaccurate points. We consider three such methods in the following sections.

### 4.4 The Perceptron "Learning" Algorithm

One way of finding the solution of a large number of homogeneous inequalities is by means of the iterative perceptron "learning" algorithm (Minsky and Papert [8]; Duda and Hart [9]). Given a set of vectors $\bar{s}_i$, this procedure is guaranteed to find a vector $\underset{\sim}{t}$ that satisfies $\bar{s}_i\cdot\underset{\sim}{t} > 0$ if such a vector exists. It even does this in a finite number of steps, provided there exists some $\epsilon$ such that $\bar{s}_i\cdot\underset{\sim}{t} > \epsilon$ for all $\bar{s}_i$ in the given set (which almost always happens when the set is finite).

The idea is to start with some non-zero vector $\underset{\sim}{t}^0$ and to test whether the inequalities are satisfied. A reasonable choice for $\underset{\sim}{t}^0$ is one of the vectors $\bar{s}_i$ or the average of all the $\underset{\sim}{s}_i$ vectors. If the inequality is not satisfied for a particular vector in the set, then the smallest adjustment is made to make the dot-product zero. (Note that this may disturb inequalities that have passed the test already). Suppose that the present estimate for the direction of the translation vector is the direction of $\underset{\sim}{t}^n$. We now test the dot-product $\bar{s}_i\cdot\underset{\sim}{t}^n$. If it is negative, we adjust our estimate of the vector $\underset{\sim}{t}$ according to the rule

$$\underset{\sim}{t}^{n+1} = \underset{\sim}{t}^n + \Delta\underset{\sim}{t} \tag{36}$$

where

$$\Delta\underset{\sim}{t} = -\frac{\underset{\sim}{s}_i\cdot\underset{\sim}{t}^n}{\underset{\sim}{s}_i\cdot\underset{\sim}{s}_i}\,\underset{\sim}{s}_i \tag{37}$$

Note that $\bar{s}_i\cdot\underset{\sim}{t}^{n+1} = 0$ and that the magnitude of $\underset{\sim}{s}_i$ does not matter. Also, the test above can be replaced with a test that checks whether $-\underset{\sim}{s}_i\cdot\underset{\sim}{t}^n$ has the same sign as $E_t$.

If the inequalities are inconsistent, that is, if the $\underset{\sim}{s}_i$ are not confined to a hemisphere, as will happen in practice due to noise, the algorithm may not converge. Furthermore, even if the algorithm converges there is no guarantee that the guess at any stage is particularly good. Several simple refinements discussed below can help in this case.

i) Random Sampling. The perceptron algorithm should not use data from picture cells scanned in some regular way, since the information from neighboring picture cells is likely to be highly correlated. Much more rapid progress is achieved by random sampling of the image region of interest. In this case there is rapid improvement in the estimate of $\underset{\sim}{t}$ while the first hundred or so picture cells are considered. Then the value is refined while the next few hundred are passed through. After a few thousand trials, the estimate is updated rarely and wanders around erratically.

ii) Partial Correction. It is likely that places where $E_x$ and $E_y$ are small do not provide information as trustworthy as do places where $E_x^2+E_y^2$ is large. We may thus hesitate to apply the full "correction" called for in the perceptron algorithm when $\underset{\sim}{s}_i\cdot\underset{\sim}{s}_i$ is small. One can simply add to the denominator in Eq. (37) a constant term $n^2$ of size commensurate with the expected noise in $\underset{\sim}{s}_i\cdot\underset{\sim}{s}_i$. The modified correction term is then

$$\Delta\underset{\sim}{t} = -\frac{\underset{\sim}{s}_i\cdot\underset{\sim}{t}^n}{\underset{\sim}{s}_i\cdot\underset{\sim}{s}_i+n^2}\,\underset{\sim}{s}_i \tag{38}$$

iii) Discarding Outliers. In applying the perceptron algorithm to real images, one must be aware that there may be vectors $\underset{\sim}{s}_i$ that are inconsistent with the preceding analysis. For example, an occluding boundary can result in arbitrary values for the estimated brightness derivatives, as can a non-functioning element in a solid-state camera. Thus it makes sense to perform a "reasonableness test" on the correction vector $\Delta\underset{\sim}{t}$. For example, if the current value of $\|\Delta\underset{\sim}{t}\|$ exceeds, say, twice the average of the previous few values, it should be discarded. (This technique may result in an incorrect final estimate if the discard rule is too stringent, however.)

iv) Terminating the Process. There does not seem to be an obvious, elegant way to end the perceptron algorithm. The following two simple procedures seem to work equally well: terminate after a fixed number of steps, or after a number of successive update vectors have magnitude $\|\Delta\underset{\sim}{t}\|$ less than some small value.

v) Choosing the Final Value. One has to decide what value to use as the final estimate of $\underset{\sim}{t}$. Rather than merely using the last position in the random stagger near the "correct" value, one may choose to use an exponentially weighted average of past values obtained using the iterative rule

$$\bar{\underset{\sim}{t}}^{n+1} = (1-\alpha)\bar{\underset{\sim}{t}}^n + \alpha\underset{\sim}{t}^n \tag{39}$$

for some small positive $\alpha$. One should not expect miracles from this added wrinkle, however.

The vector $t^n$ in the perceptron "learning" algorithm is obviously a linear combination of vectors drawn from the set $(s_i)$. Vectors in this set have directions that correspond to points in the permissible band. Now suppose that this band is very narrow. Then, to build a vector with a significant z-component one has to add many of these vectors. In order to keep the x- and y-components small, these vectors must almost come in pairs from opposite ends of the narrow band. Not surprisingly, the algorithm performs rather poorly in this situation; it is much happier with vectors sprinkled uniformly in direction over a full hemisphere.

If the FOE lies outside the field of view, then the compatible hemisphere intersects the permissible band at a significant angle. In such a case if a fraction of the s vectors lie outside the compatible hemisphere, then the estimate of the motion calculated by the perceptron method will be in error. The magnitude and direction of this error can be estimated from the sequence of corrections made by the algorithm and hence can be compensated for by a modification of the update procedure.

The perceptron algorithm as it stands is clearly sequential in nature. There is considerable interest these days in parallel methods, since these may eventually lead to "real-time" implementations. The following simple modification makes it possible for corrections to be computed simultaneously at many picture cells. Suppose each picture cell has a processor which calculates a local estimate of the vector t. Consider the process at the i-th picture cell; its initial estimate is just $t_i^0 = s_i$. Let its estimate at some later stage be $t_i^n$. The processor then computes an average of the values of its neighbors, $\bar{t}_i^n$ say, and checks whether $s_i \cdot \bar{t}_i^n > 0$. If so, it accepts this average as its new local estimate $t^{n+1}$. If the dot-product is negative, on the other hand, it adjusts $\bar{t}_i^n$ to obtain a new estimate in the traditional manner of a perceptron. (The processors corresponding to picture cells on the boundary of the image region have to do something slightly different from the rest, since they do not have a full set of neighbors). After many iterations, constraints propagate across the region and everyone pretty much agrees on a value for t. (However, there can be some differences across the region that depend on the way the average is computed).

As in so many "parallel" algorithms, we find that after the first few iterations, few updates occur and most processors have little to do. Thus the speedup to be expected is nowhere near proportional to the number of processors.

It should also be noted that in a "real-time" application, we do not expect the velocity estimates to change rapidly. Thus the previous value of the velocity is likely to be an excellent first estimate for the current value of t. This means that very few iterations will be needed to get an acceptable new value. A considerable amount of computation can be saved this way, just as it can in the computation of the optical flow (Horn and Schunck [6]).

### 4.5 Minimizing the Number of Outliers

The brightness change constraint equation can be rewritten as

$$Z = \frac{\mathbf{s}\cdot\mathbf{t}}{-E_t} = \frac{-\text{sign } E_t \; \mathbf{s}\cdot\mathbf{t}}{|E_t|} = \frac{\bar{\mathbf{s}}\cdot\mathbf{t}}{|E_t|} \qquad (40)$$

Then since Z is positive, so is $\bar{s}\cdot t$. this means that for each point in the image, the vector $\bar{s}$ lies in the hemisphere with t at its navel.

Now for real images these will typically be a small number of points for any estimate of the motion t for which $\bar{s}\cdot t < 0$. The causes of such outliers are several:

- occlusions in the scene invalidate the assumption that the brightness of a small surface patch does not change due to motion
- scene illumination gradients have the same effect
- quantization error and electronic noise can cause the sign of $E_t$ to change for $E_t$ small
- estimating derivatives by first (or low order) differences causes errors which can move a point out of the compatible hemisphere
- sampling the image (in space or time) at a rate lower than the Nyquist frequency can invalidate the brightness change constraint equation.

As we have seen, we cannot determine the actual translation vector t; only its direction can be found. Hence in the following paragraph we assume that $\|\mathbf{t}\| = 1$ and regard t as a point on the unit sphere.

For a given image sequence the number of outliers is a function of the estimated translation vector t'. With perfect data and a uniform distribution of points on the $\bar{s}$ sphere, the number of outliers is zero for $t' = t$ and increases monitonically with the magnitude of the error vector

$$\mathbf{e} = \mathbf{t} - \mathbf{t}' \qquad (41)$$

for $\|e\|$ small. For real, noisy images the relationship between the number of outliers and e is too complex to characterize simply. For example, although the number of outliers will be small for $e = 0$, there is no guarantee that it will be minimized at this point.

Given a reasonable initial estimate of t, experiments with synthetic images have shown that the value of t'which minimizes the number of outliers can be found efficiently by hill climbing. First the surface of the $\bar{s}$-sphere is tesselated in a convenient manner and the number of image points in each bin is determined. Then the number of outliers associated with a given t' can be found simply by integrating over those bins outside the great circle defined by $\bar{s}\cdot t' = 0$. Then hill-climbing can be used to find the point on the sphere where the number of outliers is minimized.

### 4.6 Minimizing the Integral of $Z^2$

In this section we assume that the depth range $Z_{max}/Z_{min}$ is limited. This will generally be the

case in robotic applications. The method discussed in this section can also be applied to images in which the background has very large Z if, as discussed in Section 4.3, these regions are excised from the image before the motion vector is calculated. We have seen that we can compute depth when the motion $\underset{\sim}{t}$ is known using Eq. (41)

$$Z = -(1/E_t)\underset{\sim}{s}\cdot\underset{\sim}{t} \qquad (42)$$

Now if we use the wrong value t' in this formula, we get the wrong depth value:

$$Z' = Z(\underset{\sim}{s}\cdot\underset{\sim}{t})/(\underset{\sim}{s}\cdot\underset{\sim}{t}'). \qquad (43)$$

We expect only positive values for Z, but the formula may give us negative values, since $\underset{\sim}{s}\cdot\underset{\sim}{t}'$ may be negative where $\underset{\sim}{s}\cdot\underset{\sim}{t}$ is positive and vice versa. More interestingly, we may obtain very large magnitudes for Z (both positive and negative), since $\underset{\sim}{s}\cdot\underset{\sim}{t}'$ may be almost zero while $\underset{\sim}{s}\cdot\underset{\sim}{t}$ is not. That is, the estimated magnitude of Z may be very large near points where $E_t \cong 0$. We may conclude that we can determine the correct value for $\underset{\sim}{t}$ by minimizing the integral of $Z^2$ over the image, that is by minimizing the quadratic form

$$\iint (1/E_t)^2(\underset{\sim}{s}\cdot\underset{\sim}{t})^2 = \underset{\sim}{t}^T\left[\iint (1/E_t^2)\underset{\sim}{s}\,\underset{\sim}{s}^T\right]\underset{\sim}{t} \qquad (44)$$

subject to the constraint $||\underset{\sim}{t}|| = 1$. The solution is the eigenvector of the real symmetric 3x3 matrix

$$M = \iint (1/E_t^2)\underset{\sim}{s}\,\underset{\sim}{s}^T \qquad (45)$$

associated with the smallest eigenvalue of M. We can prove this by minimizing the sum

$$S = \underset{\sim}{t}^T M \underset{\sim}{t} + \lambda(1 - \underset{\sim}{t}^T\underset{\sim}{t}) \qquad (46)$$

where $\lambda$ is a Lagrangian multiplier. Then

$$\frac{\partial S}{\partial \underset{\sim}{t}} = M\underset{\sim}{t} + M\underset{\sim}{t} - 2\lambda\underset{\sim}{t} = 0 \qquad (47)$$

which yields

$$M\underset{\sim}{t} = \underset{\sim}{t} \qquad (48)$$

Thus $\lambda$ is an eigenvalue of M and $\underset{\sim}{t}$ is the corresponding eigenvector. Substituting Eq. [48] into Eq. [46] gives the result $S = \lambda$. Thus $\underset{\sim}{t}^T M \underset{\sim}{t}$ is minimized by taking the smallest of the three eigenvalues of M for $\lambda$. To minimize problems due to noise, we might add a small positive constant to $E^2$ commensurate with the expected noise in $E^2$. That is, we take as our solution the eigenvector of

$$\iint \frac{1}{E_t^2 + n^2}\,\underset{\sim}{s}\,\underset{\sim}{s}^T \qquad (49)$$

associated with the smallest eigenvalue.

If $\underset{\sim}{e}$ is an eigenvector so is $-\underset{\sim}{e}$. But we want Z to be positive. Rather than test this condition at a single point, we compute an average like

$$\underset{\sim}{\bar{s}}_0 = -\iint (1/E_t)\underset{\sim}{s} \quad \text{or} \quad \underset{\sim}{\bar{s}}_0 = -\iint E_t/(E_t^2+n^2)\underset{\sim}{s} \qquad (50)$$

and check whether $\underset{\sim}{\bar{s}}_0\cdot\underset{\sim}{t} > 0 \qquad (51)$

If it is not, we simply change the sign of the solution.

As before, we may choose to weight the integral of Eq. (44) according to some measure of how trustworthy the data from each picture cell is.

The method presented in this section produces an estimate to the translation vector $\underset{\sim}{t}$ in closed form and with high accuracy. Of course, a cubic must be solved to obtain the eigenvalues--but there is an analytic method for doing that. The corresponding eigenvectors can then be found by taking cross-products of two rows of a 3x3 matrix.

The preceding method of calculating $\underset{\sim}{t}$ has another justification. From Eq. [38] we know that $\underset{\sim}{s}\cdot\underset{\sim}{t} \cong 0$ whenever $E_t \cong 0$ (again ignoring background points). Thus we are basically looking for a vector $\underset{\sim}{t}$ that makes $\underset{\sim}{s}\cdot\underset{\sim}{t} = 0$ whenever $E_t \cong 0$. The points where the time derivatives are small provide most constraint, as already discussed. We could try to minimize something like.

$$\iint_C (\underset{\sim}{s}\cdot\underset{\sim}{t})^2 \qquad (52)$$

where C is the set of image points where $E_t \cong 0$. Rather than use a strict cut-off, we may consider a weighting scheme in an integral like

$$\iint w(\underset{\sim}{s}\cdot\underset{\sim}{t})^2 \qquad (53)$$

over the whole image where the weighting function is chosen to emphasize points where $E_t \cong 0$. A reasonable choice, $w = 1/(E^2+n^2)$, leads to the integral given in Eq. (44). The eigenvector corresponding to the smallest eigenvalue is a normal of the plane that best fits the weighted set of points.

In general, an eigenvector of a matrix M may be obtained by solving a homogeneous set of equations whose coefficient matrix is $(M-\lambda I)$, for a given eigenvalue $\lambda$. The solution will clearly be orthogonal to each row of the matrix $(M-\lambda I)$. Such a solution may be found then by taking cross-products of two rows of this matrix. There are three ways to form such cross-products. Due to inaccuracies in the calculation, the three cross-products may not be exactly parallel. It helps then to add them, after making sure that they all have the same sense. This provides one with a more accurate answer.

In some cases it may not be necessary to explicitly factor a cubic to determine the smallest eigenvalue. With good data, the smallest eigenvalue is a lot smaller than the other two. In this case there is little difference between M and $(M-\lambda I)$. We can get good approximations of the eigenvector associated with the smallest eigenvalue by taking cross-products of the rows of M rather than $(M-\lambda I)$. This obviates the need to solve the cubic to determine the eigenvalues. We have found that this simplified method works well.

Figure 8 shows a scatter plot of positions on the unit sphere for $\underset{\sim}{t}$ recovered from noisy synthetic data. Each estimate is based on brightness gradients at 200 picture cells with 1% noise. Note the elongation of the cluster of points in a direction parallel to the optical axis.

## 5. Conclusions

We have developed methods for recovering motion directly from brightness derivatives in an image subject to purely translation motion or to arbitrary motion with known rotation. We have tested these methods on synthetic image data and, to a limited extent, on real image sequences. The

methods exhibit different trade-offs between accuracy, noise-sensitivity and computational expense. Detailed evaluation of the relative merits of these methods under a variety of scenes and viewing conditions is underway.

## 6. Acknowledgements

This research was supported by the National Science Foundation under Grant No. DMC85-11966. Additional support was provided by NASA (Grant No. GSFC 5-1162), the Veteran's Administration, and the Pacific International Center for High Technology Research

The authors would like to thank the other members of the Waikiki Surfing and Computer Vision Society, and especially Shahriar Negahdaripour, for their contributions to this paper.

## 7. References

[1] Horn, B.K.P. and E.J. Weldon, Jr., "Robust Direct Methods for Recovering Motion", International Journal of Computer Vision, to appear, 1987.

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

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

[4] Negahdaripour, S., Horn, B.K.P., "Direct Passive Navigation", IEEE Trans. Pattern Analysis and Machine Intelligence, 1986, in press.

[5] Waxman, A.M., Ullman, S., "Surface Structure and 3-D Motion from Image Flow: A Kinematic Analysis", CAR-TR-24, Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park, MD, October 1983.

[6] Horn, B.K.P., Schunck, B.G., "Determining Optical Flow, "Artificial Intelligence, Vol. 17, pp. 185-203, 1981.

[7] Longuet-Higgins, H.C., Prazdny, K., "The Interpretation of a Moving Retinal Image", Proc. of Royal Society of London, Series B, Vol. 208, pp. 385-397, 1980.

[8] Minsky, M. and S. Papert, "Perceptrons: An Introduction to Computational Geometry", MIT Press, Cambridge, MA, 1969.

[9] Duda, R.O., and P.E. Hart, "Pattern Classification and Scene Analysis", John Wiley, New York, 1973.

[10] Negahdaripour, S., "Direct Mehtods for Structure from Motion", Ph.D. Thesis, Mechanical Engineering, MIT, 1986.

[11] Wolf, P.R., "Elements of Photogrammetry", McGraw-Hill, New York, 1983.

[12] Gosh, S.K., "Theory of Stereophotogrammetry", Ohio State University Bookstores, 1972.

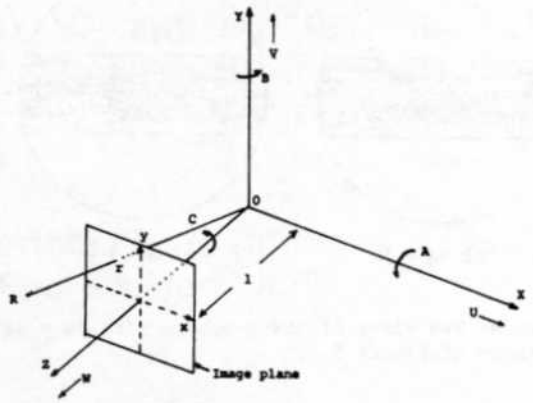


Figure 1 The viewer-centered coordinate system.

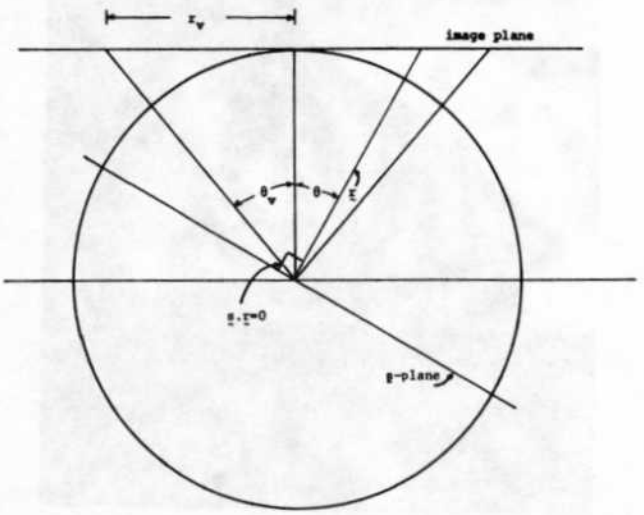


Figure 2 A cross-section through the g-sphere defined by the image point r.

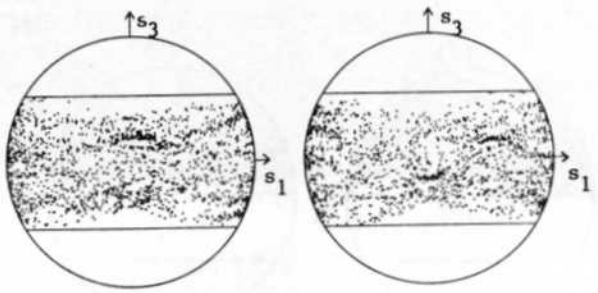


Figure 3 Two views of the g-sphere with $\theta_v = 33°$ for image of Figure 5.

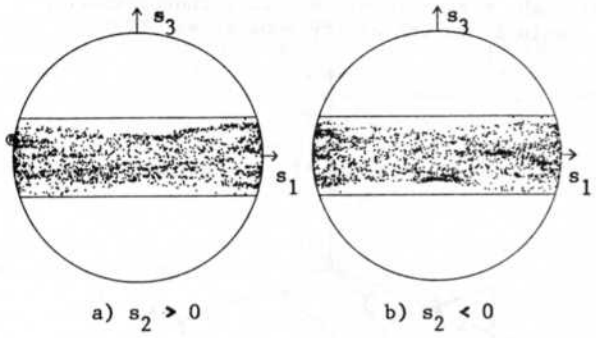


Figure 4 Two views of the $\underset{\sim}{s}$-sphere with $\theta_v = 19°$ for image of Figure 5.

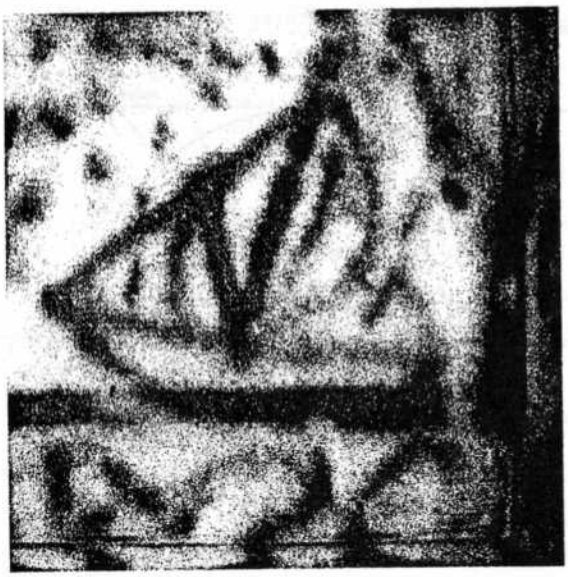

Figure 5 Planar band-limited scene; $\theta_v = 33°$; typical depth: 1.3 m. (Untitled, spray paint on plywood, by the authors, 1986)

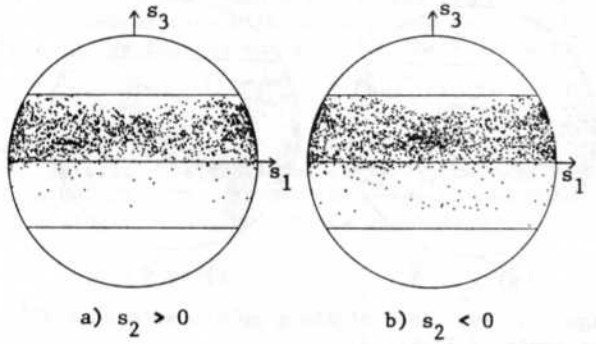


Figure 6 $\bar{\underset{\sim}{s}}$-sphere for $\underset{\sim}{s}$-sphere of Figure 3 with $\underset{\sim}{t} = (0,0,2)^T$cm. FOE near image center.

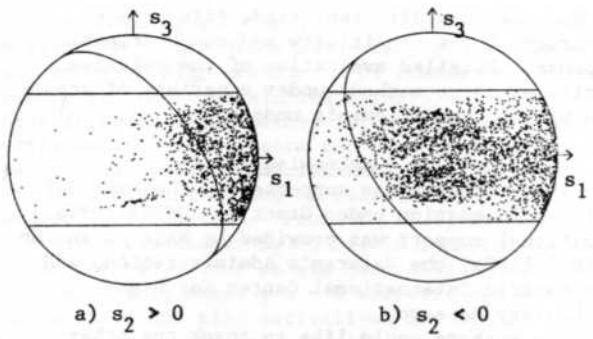


Figure 7 $\bar{\underset{\sim}{s}}$-sphere for $\underset{\sim}{s}$-sphere of Figure 3 with $\underset{\sim}{t} = (2,-\sqrt{2},\sqrt{2})^T$cm. FOE well outside image (45°).

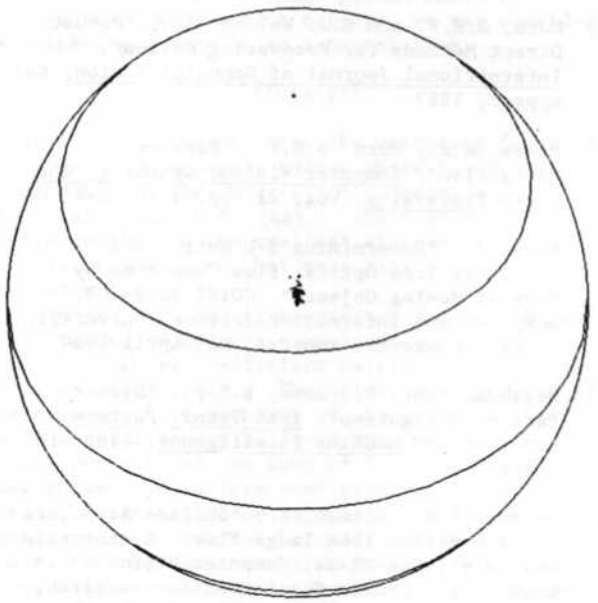

Figure 8 Plot of several noisy estimates of the translation vector $\underset{\sim}{t}$ on the $\underset{\sim}{s}$-sphere (200 pixels/estimate, 1% noise) calculated by the min $Z^2$ method.