A Fast Method to Estimate Sensor Translation

*V. Sundareswaran**

Courant Institute, New York University, New York, NY 10012

Abstract. *An important problem in visual motion analysis is to determine the parameters of egomotion. We present a simple, fast method that computes the translational motion of a sensor that is generating a sequence of images. This procedure computes a scalar function from the optical flow field induced on the image plane due to the motion of the sensor and uses the norm of this function as an error measure. Appropriate values of the parameters used in the computation of the scalar function yield zero error; this observation is used to locate the Focus of Expansion which is directly related to the translational motion.*

1 Introduction

We consider the motion of a sensor in a rigid, static environment. The motion produces a sequence of images containing the changing scene. We want to estimate the motion of the sensor, given the optical flow fields computed from the sequence. We model the motion using a translational velocity T and a rotational velocity ω. These are the instantaneous motion parameters.

Many procedures exist to compute the optical flow field [1,4]. Also, several methods have been proposed to compute the motion parameters from the optical flow field. One feature of most of these methods is that they operate locally. Recovering structure, which is contained in local information, seems to be the motivation for preferring local methods. However, the motion parameters are not local and they are better estimated by employing global techniques. In addition, using more data usually results in better performance in the presence of noise. Non-local algorithms are given in [3] and [8], and more recently, in [6]. The algorithm presented in [3] requires search over grid points on a unit sphere. The method of Prazdny [8] is based on a non-linear minimization. Faster methods have been presented recently [7,10]. Though all these methods work well on noiseless flow fields, there is insufficient data about their performance on real images. The work in this paper has been motivated by the observation that making certain approximations to an exact procedure gives a method that produces robust results from real data.

The algorithm presented here determines the location of the focus of expansion (FOE) which is simply the projection of the translation vector T on the imaging plane. It is well known that once the FOE is located, the rotational parameters can be computed from the optical flow equations [2]. Alternative methods to directly compute the rotational parameters from the flow field have also been proposed [11]. We begin by reviewing the flow equations, then describe the algorithm and present experimental results.

2 The flow equations

We consider the case of motion of a sensor in a static environment. We choose the coordinate system to be centered at the sensor which uses perspective projection for

* Supported under Air Force contract F33615-89-C-1087 reference 87-02-PMRE. The author wishes to thank Bob Hummel for his guidance.

imaging onto a planar image surface (Fig. 1). The sensor moves with a translational velocity of $T = (v_1, v_2, v_3)$ and an angular velocity of $\omega = (\omega_1, \omega_2, \omega_3)$.

The transformation from spatial coordinates to the image coordinates is given by the equations

$$x = fX/Z, \quad y = fY/Z$$

where $(X, Y, Z) = (X(x, y), Y(x, y), Z(x, y))$ is the position of the point in three-space that is imaged at (x, y) and f is the focal length. The optical flow $V = (u, v)$ at the image point (x, y) is easily obtained [2,5,9]:

$$\begin{aligned} u(x,y) &= \tfrac{1}{Z(x,y)}[-fv_1 + xv_3] + \omega_1\left[\tfrac{xy}{f}\right] - \omega_2\left[f + \tfrac{x^2}{f}\right] + \omega_3 y, \\ v(x,y) &= \tfrac{1}{Z(x,y)}[-fv_2 + yv_3] + \omega_1\left[f + \tfrac{y^2}{f}\right] - \omega_2\left[\tfrac{xy}{f}\right] - \omega_3 x. \end{aligned} \tag{1}$$

Here, $u(x, y)$ and $v(x, y)$ are the x and y components of the optical flow field $V(x, y)$. In this context, we are interested in determining the location $(\tau = fv_1/v_3, \eta = fv_2/v_3)$ which is nothing but the projection of the translational velocity T onto the image plane. This location is referred to as the Focus of Expansion (FOE).

Looking at Eqn. 1, we note that the vector flow field $V(x, y)$ is simply the sum of the vector field $V_v(x, y)$ arising from the translation T and the vector field $V_\omega(x, y)$ due to the rotation ω:

$$V(x, y) = V_v(x, y) + V_\omega(x, y).$$

3 Algorithm Description

The observation behind the algorithm is that a certain circular component computed from the flow field by choosing a center (x_0, y_0) is a scalar function whose norm is quadratic in the two variables x_0 and y_0. The norm is zero (in the absence of noise) at the FOE. This procedure will be referred to as the Norm of the Circular Component (NCC) algorithm.

3.1 The circular component

For each candidate (x_0, y_0), we consider the circular component of the flow field about (x_0, y_0) defined by:

$$U_{(x_0,y_0)}(x, y) = V(x, y) \cdot (-y + y_0, x - x_0). \tag{2}$$

Note that this is nothing but the projection onto concentric circles whose center is located at (x_0, y_0). Since $V = V_v + V_\omega$, we further define

$$\begin{aligned} U^v_{(x_0,y_0)}(x, y) &= V_v(x, y) \cdot (-y + y_0, x - x_0), \\ U^\omega_{(x_0,y_0)}(x, y) &= V_\omega(x, y) \cdot (-y + y_0, x - x_0), \end{aligned}$$

so that

$$U_{(x_0,y_0)}(x, y) = U^v_{(x_0,y_0)}(x, y) + U^\omega_{(x_0,y_0)}(x, y)$$

where, denoting $\rho(x, y) = 1/Z(x, y)$,

$$U^v_{(x_0,y_0)}(x, y) = v_3\rho(x, y) \cdot [(y_0 - \eta)x + (-x_0 + \tau)y + \eta x_0 - \tau y_0], \tag{3}$$

and

$$U^{\omega}_{(x_0,y_0)} = \omega_1 \left[-\frac{x_0}{f}y^2 + \frac{y_0}{f}xy + fx - fx_0\right] + \omega_2 \left[-\frac{y_0}{f}x^2 + \frac{x_0}{f}xy + fy - fy_0\right] + \omega_3 \left[-y^2 - x^2 + x_0x + y_0y\right]. \quad (4)$$

At the focus of expansion, when $(x_0, y_0) = (\tau, \eta)$,

$$U^{v}_{(x_0,y_0)}(x,y) = v_3\rho(x,y) \cdot \begin{bmatrix} x-\tau \\ y-\eta \end{bmatrix} \cdot (-y+\eta, x-\tau) = 0 \quad (5)$$

so that $U_{(x_0,y_0)} = U^{\omega}_{(x_0,y_0)}$ for $(x_0, y_0) = (\tau, \eta)$. Eqn. 5 is merely a result of the radial structure of the translational component of the flow field. In other words, pure translation produces a field that is orthogonal to concentric circles drawn with the FOE as the center. Observations about the quadratic nature of $U^{\omega}_{(x_0,y_0)}$ (Eqn. 4) lead to the convolution and subspace projection methods described in [6]. Here, we obtain a method that is approximate but is quick and robust.

To this end, we define an error function $E(x_0, y_0)$ as the norm of $U_{(x_0,y_0)}(x,y)$:

$$E(x_0, y_0) = \|U_{(x_0,y_0)}(x,y)\|^2. \quad (6)$$

The important observation is that $U_{(x_0,y_0)}(x,y)$ is *linear* in the parameters x_0 and y_0. As a result, the norm defined in Eqn. 6 will be *quadratic* in x_0 and y_0. That is, $E(x_0, y_0)$ will be a quadratic polynomial in x_0 and y_0. The minimum of this quadratic surface is purported to occur at the Focus of Expansion (FOE). We will justify this claim shortly. But first, if the claim is correct, we have a simple algorithm that we describe now.

3.2 The NCC algorithm

The first step is to choose six sets of values for (x_0, y_0) in a non-degenerate configuration (in this case, non-collinear). Next, for each of these candidates, compute the circular component and define $E(x_0, y_0)$ to be the norm of the circular component (NCC). In a discrete setting, the error value is simply the sum of the squares of the circular component values. Note that this can be done even in the case of a sparse flow field. The error function values at these six points completely define the error surface because of its quadratic nature and so the location of the minimum can be found using a closed-form expression. That location is the computed FOE.

Let us now examine the claim about the minimum being at the location of the FOE. Note that the function $U_{(x_0,y_0)}(x,y)$ is made up of two parts; one is the translational part shown in Eqn. 3, and the other is the rotational part (Eqn. 4). The translational part $U^{v}_{(x_0,y_0)}(x,y)$ vanishes at the FOE, as shown in Eqn. 5, and it is non-zero elsewhere. Thus, the norm $\|U^{v}_{(x_0,y_0)}(x,y)\|^2$ is positive quadratic with minimum (equal to zero) at the FOE. This is no longer true once we add the rotational part. However, as long as the contribution from the rotational part is small compared to that from the translational part, we can *approximate* the behavior of $\|U^{v}_{(x_0,y_0)}(x,y)\|^2$ by $\|U_{(x_0,y_0)}(x,y)\|^2$.

The method is exact for pure translation and is approximate when the rotation is small compared to the translation or when the depth of objects is small (i.e., high $\rho(x,y)$) as would be the case in indoor situations. Also, there is no apparent reason for this method to fail in the case where a planar surface occupies the whole field of view. Previous methods [7,10] are known to fail in such a case. Indeed, in two experiments reported here, a large portion of the view contains a planar surface. In all experiments done with synthetic as well as actual data, this algorithm performs well. We present results from actual image sequences here.

4 Experiments

For all the sequences used in the experiments, the flow field was computed using an implementation of Anandan's algorithm [1]. The dense flow field thus obtained (on a 128 by 128 grid) is used as input to the NCC algorithm. The execution time per frame is on an average less than 0.45 seconds for a casual implementation on a SUN Sparcstation-2.

The helicopter sequences, provided by NASA, consist of frames shot from a moving helicopter that is flying over a runway. For the straight line motion, the helicopter has a predominantly forward motion, with little rotation. The turning flight motion has considerable rotation. The results of applying the circular component algorithm to these sequences are shown in Figure 2 for ten frames (nine flow fields). This is an angular error plot, the angular error being the angle between the actual and computed directions of translation. The errors are below 6 degrees for all the frames of the straight flight sequence. Notice the deterioration in performance towards the end of the turning flight sequence due to the high rotation (about 0.15 rads/sec).

The results from a third sequence (titled *ridge*, courtesy David Heeger) are shown in Figure 2. Only frames 10 through 23 are shown because the actual translation data was readily available only for these frames. In this sequence, the FOEs are located relatively high above the optical axis. Such sequences are known to be hard for motion parameter estimation because of the confounding effect between the translational and rotational parameters (see the discussion in [6]). The algorithm presented here performs extremely well, in spite of this adverse situation.

5 Conclusions

In most practical situations, the motion is predominantly translational. However, even in situations where only translation is intended, rotation manifests due to imperfections in the terrain on which the camera vehicle is traveling or due to other vibrations in the vehicle. Algorithms that assume pure translation will break down under such circumstances if they are sensitive to such deviations. However, the algorithm described here seems to tolerate small amounts of rotation. So, it can be expected to work well under the real translational situations and for indoor motion where the small depth values make the translational part dominant.

In addition to the above situations, the method described here could also be used to provide a quick initial guess for more complicated procedures that are designed to work in the presence of large rotational values.

References

1. P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *International Journal of Computer Vision*, 2:283–310, 1989.
2. D. Heeger and A Jepson. Subspace methods for recovering rigid motion I: Algorithm and implementation. Research in Biological and Computational Vision Tech Rep RBCV-TR-90-35, University of Toronto.
3. D. Heeger and A. Jepson. Simple method for computing 3d motion and depth. In *Proceedings of the 3rd International Conference on Computer Vision*, pages 96–100, Osaka, Japan, December 1990.
4. David J. Heeger. Optical flow using spatiotemporal filters. *International Journal of Computer Vision*, 1:279–302, 1988.

5. B.K.P Horn. *Robot Vision.* The MIT Press, 1987.
6. Robert Hummel and V. Sundareswaran. Motion parameter estimation from global flow field data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.
7. A. Jepson and D. Heeger. A fast subspace algorithm for recovering rigid motion. In *IEEE Workshop on Visual Motion*, Princeton, New Jersey, Oct 1991.
8. K.Prazdny. Determining the instantaneous direction of motion from optical flow generated by a curvilinearly moving observer. *Computer Vision, Graphics and Image Processing*, 17:238–248, 1981.
9. H.C. Longuet-Higgins and K. Prazdny. The interpretation of a moving retinal image. *Proc. Royal Soc. Lond. B*, 208:385–397, 1980.
10. V. Sundareswaran. Egomotion from global flow field data. In *IEEE Workshop on Visual Motion*, Princeton, New Jersey, Oct 1991.
11. V. Sundareswaran and R. Hummel. Motion parameter estimation using the curl of the flow field. In *Eighth Israeli Conference on AI and Computer Vision*, Tel-Aviv, Dec 1991.

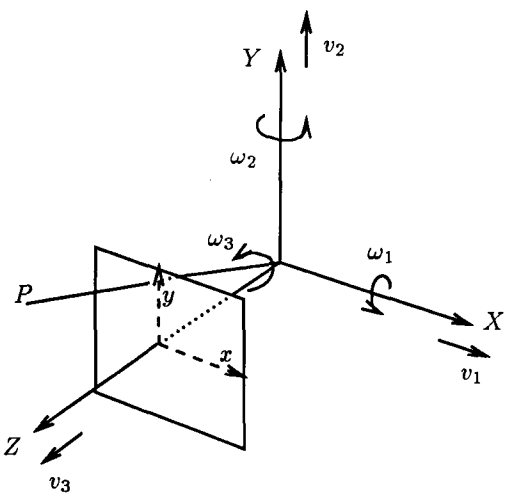

Fig. 1. The coordinate systems and the motion parameters

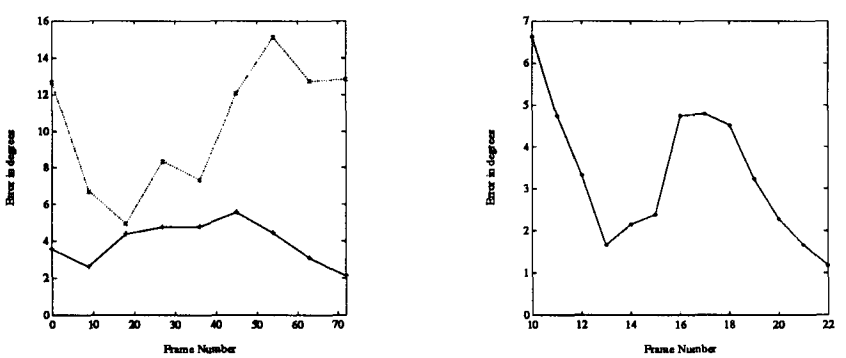

Fig. 2. Angular error plots for the helicopter sequences(left: straight line flight in solid line and turning flight in dotted line) and the ridge sequence (right)