VELOCITY ESTIMATION FROM IMAGE SEQUENCES WITH SECOND ORDER DIFFERENTIAL OPERATORS

OLEH TRETIAK* - LOUIS PASTOR**

* Drexel University, Philadelphia, PA
** Departamento de Automatica, ETSII - UPM, Madrid, Spain

ABSTRACT

An algorithm is presented that estimates velocities of objects from second order spatial and spatio-temporal derivatives. Images are first smoothed to increase the signal-to-noise ratio and to assure continuity of the intensity function. Experimental results show that the algorithm provides good estimates even when the conditions are not ideal: rotation and nonuniform background are tolerated when present in small amounts.

1 INTRODUCTION

Methods for motion estimation can be classified as matching algorithms that deduce motion by comparing locations of features on succesive frames, and local algorithms that estimate velocities from spatial and temporal variations of intensity. A number of these non-matching procedures are based on the so-called 'optical flow constraint'[1,2]

$$I_t = - I_x V_x - I_y V_y \tag{1}$$

Where I(x,y,t) is the image intensity as a function of two spatial coordinates and time, I_x, I_y I_t are partial derivatives, and V_x, V_y are velocity components. Partial derivatives can be approximated by differences. This equation only provides a constraint between the two components of velocity, and another relation is necessary to obtain unambiguous values of the velocity vector. Some investigators[3,4] computed only one velocity component. Other studies used a clustering approach[4,5] (a modified Hough transform) to both estimate velocity and to segment the image into static and moving areas. Powerful algorithms[2,6] can be developed if one assumes that velocity varies smoothly over the image. Nagel[7] uses quadratic approximations of the intensity function, and matches these over succesive images. The matching idea[8] was also applied to directional derivatives of the luminance over regions in the image. Another family of 'incremental' algorithms is based on Kalman filter theory[9,10,11].

2 THEORETICAL BASIS

Let L(u,v) be a function with continuous second derivatives, and assume that

$$I(x,y,t) = L(x-V_x t, y-V_y t). \tag{2}$$

More complex situations, such as acceleration or rotary motion can be handled by using appropriate functions u(x,y,t) and v(x,y,t). In our case,

$$I_t = - L_u V_x - L_v V_y. \tag{3}$$

Differentiating this with respect to x and y, and noting that the partial derivateves I_x, I_y, I_{xx}, etc are equal to L_u, L_v, L_{uu}, etc, respectively, we get

$$I_{xt} = - I_{xx} V_x - I_{xy} V_y \tag{4.1}$$

$$I_{yt} = - I_{xy} V_x - I_{yy} V_y. \tag{4.2}$$

This is a system of two equations in the two unknowns V_x, V_y and can be solved at each point where the hessian matrix

$$H = \begin{vmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{vmatrix} \tag{5}$$

is nonsingular. Whenever this is the case, the velocity can be computed from

$$\begin{vmatrix} V_x \\ V_y \end{vmatrix} = H^{-1} \begin{vmatrix} I_{xt} \\ I_{yt} \end{vmatrix} \tag{6}$$

Note that the determinant of the hessian is the gaussian curvature of the 'surface' I(x,y,t) at fixed t. It is equal to the product of the two principal curvatures at a point, giving an idea of 'distinctiveness' of a given point. It provides important information, since the points that have large gaussian curvature are those that have large contrast in all directions, and are good candidates for velocity estimation.

3 ALGORITHM

Practical implementation of an algorithm that uses equation (6) for velocity estimation has to take two points into consideration:

CH2046-1/84/0000/0016$01.00©1984 IEEE

1. Equation (6) was derived assuming that the image and its partial derivatives were continuous. Real images are discontinuous in regions of sharp contrast, like object boundaries, shadows, etc. This is specially true for synthetic images, where no blurring is introduced by the scanner. Smoothing the image with a low-pass filter insures the continuity of the intensity function, and also increases the signal to noise ratio.

2. In a digital computer the image is stored as an array of points, rather than a function of continuous variables. Therefore partial derivatives have to be approximated by differences.

The algorithm developed by us can be decomposed into the following steps:

1. Image Smoothing. Smoothing was performed with succesive sliding averages. In each smoothing stage every point in the image was replaced by the average of the intensity function over the points inside a square centered on the point. Different smoothing impulse response shapes were obtained by performing the smoothing stage different numbers of times: after two stages, the impulse response has a pyramidal shape, after three, a paraboloidal bell shape, etc. In our experiments, images were digiized to eight bits, but it was necessary to retain fractional pixel brightness values in the smoothing process.

   The filter described above performs spatial smoothing. It is also possible to perform temporal smoothing by averaging consecutive frames within a sequence. The main advantage of time averaging is that the width of the resulting spread depends on the velocity of the mooving object, and permits better estimates for large speeds. Its main disadvantage are greater computational and storage requirements. No temporal smoothing was performed in our experiments.

2. Partial Derivative Calculation. Partial derivatives were approximated with the following central differences:

$$I_{xx}(x,y,t) = [I(x+2,y,t) - 2I(x,y,t) + I(x-2,y,t)]/4 \quad (6)$$

$$I_{xy}(x,y,t) = [I(x+1,y+1,t) - I(x-1,y+1,t) - I(x+1,y-1,t) + I(x-1,y-1,t)]/4 \quad (7)$$

$$I_{xt}(x,y,t) = [I(x+1,y,t+1) - I(x+1,y,t-1) - I(x-1,y,t+1) + I(x-1,y,t-1)]/4 \quad (8)$$

   The expressions for I_{yy} and I_{yt} are similar.

   It is possible to analyze the error introduced by these approximations by expanding the image function in a Taylor series arount the point (x,y,t). The error terms depend on intensity function derivatives of order four or higher[12].

   We found that the the difference formulas had a substantial effect on the accuracy of the velocity estimates, and best results were obtained with the above formulas.

3. Velocity Estimate Calculation. Once the second derivatives of the intensity function have been found, equation (6) gives an estimate of the interframe displacement (whenever the gaussian curvature is not equal to zero).

4. Velocity Estimate Smoothing. The accuracy of the estimates obtained in step 3 depends on many factors, such as noise, local properties of the image, the actual speed, etc. Averaging estimates over small neighborhoods is a simple way to reduce errors. Points with small values of gaussian curvature were excluded from the averages. It can be shown that the determinant of H computed from the given second differences is an unbiased estimate of gaussian curvature if the noise is white and of zero mean[12]. However, since the determinant appears in the denominator of equation (6), noise will have a larger effect if the gaussian curvature is small.

4 RESULTS

Motion was simulated in two ways: by performing translations of a single image with computer programs (synthetic motion), or by digitizing sequences of objects mooved manually (real motion). Most of the results that we will report were obtained in the vicinity of a corner of a rectangular object mooving over uniform and nonuniform backgrounds. The results obtained with synthetic and real motion trials show that the accuracy of the velocity estimate depends on several factors:

1. External factors:

   1.1 Magnitude of the interframe displacements: smaller displacements can be measured more accurately.

   1.2 Non-ideal conditions, such as rotary motion, nonuniform background, acceleration, noise, etc.

2. Internal factors:

   2.1 Filter parameters. In general, the wider the filter impulse response, the smaller the errors. This is especially true for large speeds.

   2.2 The gaussian curvature at a point. Usually, the best estimates are obtained at points with large gaussian curvature, as discussed in section 3. Reference [12] contains more details on this matter.

Figures 1 and 2 illustrate some examples of the the dependence of the average speed error and velocity standard deviation on the width of the filter impulse response and on the threshold value of gaussian curvaure. It can be seen that if the filter impulse response width is large enough, when the threshold is set to reject points with low gaussian curvature, the algorithm produces accurate velocity estimates.

5 CONCLUSIONS AND DISCUSSION

The algorithm presented provides a simple and reliable method for velocity estimation when the interframe displacement is small. Our algorithm seems to be best suited for estimating small velocities, ones that lead to interframe displacements of only a few pixels. Note that the algorithms are capable of measuring velocities that correspond to interframe displacements amounting to fractions of pixels.

This algorithm is a member of a family of techniques that are based on both first and second partial derivatives of images. Note that the information about the velocity contained in equation (1) is independent of that in equations (4). By combining these we obtain three equations in two unknowns (the velocity components) and the velocity may be found by computing the pseudoinverse. A more geometrically intuitive approach is to evaluate the directional derivative of the temporal rate of change of image intensity (1) in the direction of the gradient or at right angles to it. This leads to

$$(I_x I_{xx} + I_y I_{xy})V_x + (I_x I_{xy} + I_y I_{yy})V_y = - I_x I_{xt} - I_y I_{yt} \qquad (9)$$

and

$$(I_y I_{xx} - I_x I_{xy})V_x - (I_y I_{xy} - I_x I_{yy})V_y = - I_y I_{xt} + I_x I_{yt} \qquad (10)$$

respectively. Either (9) or (10) may be combined with (1) to produce two equations for the velocity components. Note that the determinants of these groups of equations are different than that of (4) so that they may lead to valid velocity estimates even for points where the gaussian curvature is equal to zero. In general, the techniques we have described lead to a redundant system of equations and the redundancy may be exploited to design more flexible and/or robust algorithms for velocity estimation.

The algorithm may be applicable to the measurement of object displacement between similar images, such as stereo pairs.

Further research is required to explore the application of these notions to the estimation of large speeds, image segmentation, and scene analysis.

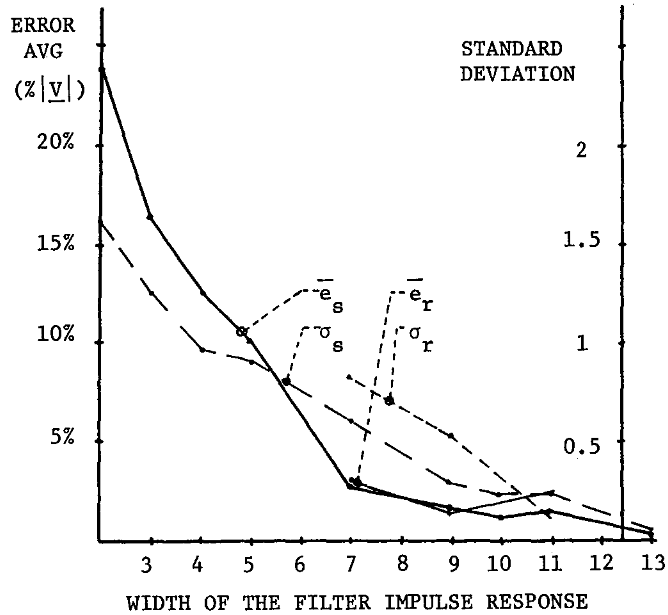

Figure 1. Dependence of error on filter impulse response width. Gaussian curvature threshold is 10% of maximum gaussian curvature. Synthetic motion, velocity (1,1) and (1,-1) pixels, $\bar{e}_s$ - average speed error, σ_s - velocity standard deviation, Real motion, velocity (-1,0), $\bar{e}_r$ - average speed error, σ_r - speed standard deviation.

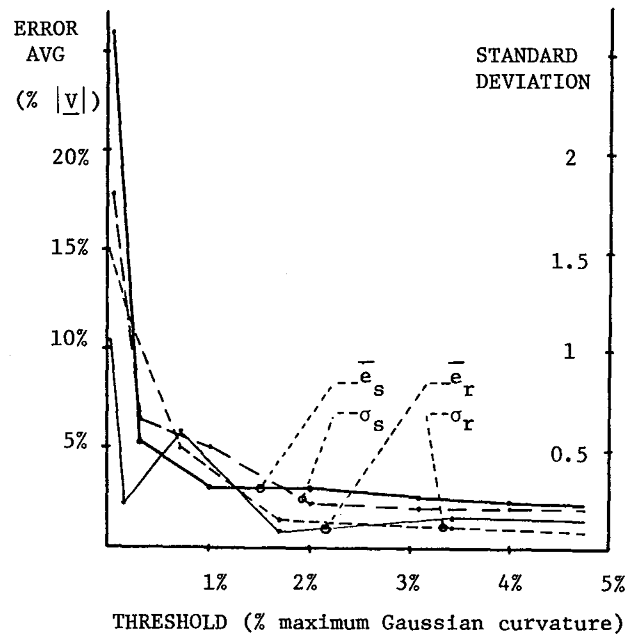

Figure 2. Dependence of error on gaussian curvature threshold. Three stages of 3x3 pexel smoothing were used (filter impulse response width is 7 pixels). Synthetic motion, velocity (1,1) and (1,-1) piexles, $\bar{e}_s$ - average speed error, σ_s - velocity standard deviation.

ACKNOWLEDGEMENT

This work was supported, in part, by NIH Grant 3P41RR01638 - 01S1 SSS.

REFERENCES

[1] Ballard D.M., Brown C.M., Computer Vision, Prentice Hall, 1982.
[2] Horn B.K.P., Schunck B.G., Determining optical flow, Artificial Intelligence, 17: 185-203 (May 1981).
[3] Limb J.O., Murphy J.A., Estimating the velocity of moving images in television signals, Computer Graphics and Image Processing, 4: 311-327 (1975).
[4] Fennema C.L., Thompson, W.B., Velocity determination in scenes containing several moving objects, Computer Graphics and Image Processing, 9: 377-381 (April 1979).
[5] Thompson W.B., Combining motion and contrast for segmentation, Technical Report 79-7, Computer Science Department, University of Minnesota, (March 1979).
[6] Hildreth E.C., The computation of the velocity field, to appear in the Proceedings of the Royal Society London B.
[7] Nagel H.H., Displacement vectors derived from second order intensity variations, Computer Graphics and Image Processing, 12: 85-117 (Jan. 1983).
[8] Paquin, R., Dubois, B., A spatiotemporal method for estimating the displacement field in time varying imagery, Computer Vision, Graphics, and Image Processing, 12: 205-221 (Feb. 1983).
[9] Acharaya R.S., Dynamic scene model: an estimation theory approach, Proc. 1982 IEEE Com. Soc. Conf. on Pattern Recognition and Image Processing, pp. 126-127 (June 1982).
[10] Stuller J.A., Krishnamurthy G., Kalman filter formulation of low-level television image motion estimation, Computer Vision, Graphics and Image Processing, 12: 169-204 (Feb. 1983).
[11] Stuller J.A., Paik H.W., Pel-recursive television image motion estimation using the Kalman filter, Proc. of MEDCOMP-82, IEEE Computer Society Press, (Sept. 1982).
[12] Pastor L., Velocity estimation from image sequences with second order differential operators, MS Thesis, Drexel University (1983).