ESTIMATION OF MOTION BOUNDARY LOCATION AND OPTICAL FLOW USING DYNAMIC PROGRAMMING

Xenophon Papademetris and Peter N. Belhumeur

Dept. of Electrical Engineering
Yale University, New Haven, CT 06520-8267
Phone: 203-432-4249 , Fax: 203-432-7481
e-mail: belhumeur@ledoux.eng.yale.edu

ABSTRACT

We present a new method for the estimation of optical flow which uses a dynamic programming based algorithm to simultaneously detect the presence of motion boundaries and to estimate optical flow. This allows for more accurate estimation of the motion field near discontinuities. The results compare favorably with those produced by other methods.

1. INTRODUCTION

The accurate measurement of optical flow or image velocity is important in many computer vision applications such as structure from motion and figure-ground segmentation.The traditional methods for optical flow estimation (see review by Barron et al [1] for a more detailed review and evaluation) do not account for the presence of motion boundaries in the scene which results in unreliable flow estimates close to object boundaries and makes the output unsuitable for use in structure for motion determination. The proposed algorithm explicitly detects the location of these boundaries and uses them to segment the scene into homogeneous regions which allows for more accurate optical flow estimation and scene segmentation.

2. PREVIOUS WORK

Differential methods compute compute flow by using spatio-temporal derivatives. The basic equation in optical flow calculation known as *the gradient constraint equation* is: $I_t + uI_x + vI_y = 0$. This equation has two unknowns - the velocity components (u, v) and further constraints are therefore necessary. Horn and Schunck add a global first order smoothness constraint term [2] whereas Lucas and Kanade [3] implement a local smoothness constraint by using a least squares fit in a small window assuming a translational model. Other approaches include region based techniques [4], energy-based methods [5] and more recently [6], and phase based methods [7]. All of these techniques however do not account explicitly for the possibility of there being motion boundaries which would directly violate the smoothness constraints used.

In general the local methods can be made more accurate by increasing the window size, as long one is sufficiently far away from the motion boundaries, near which the performance deteriorates. The size of the window is a trade-off between accurate estimation and reduction of the likelihood of it including a boundary. Szeliski [8] uses a quad-tree technique to decompose the image into square domains. In each domain an affine model is applied and if the residual error is high this domain is subdivided into four until the model fits the data within a sufficient degree of accuracy.

Global methods, as suggested by Horn [9], can be improved by suspending the smoothing on a boundary and a first step was taken by Nagel [10] where no smoothing is applied across step intensity edges. Finding these boundaries is a difficult optimization problem, Schnörr [11] assumes some prior knowledge of the location of these and deforms the boundaries to minimize an error functional.

3. OPTIMAL OPTICAL FLOW ESTIMATION IN A NARROW BAND

Consider the scene shown in Figure 1 which consists of a house surrounded by grass as seen by a camera moving from left to right (i.e. to it the house is moving to the left). If one was to concentrate only on a small horizontal band along the image, it can be seen, from a motion perspective, that it consists of three regions (the house is the middle region) which have smoothly varying velocity. An optimal estimate of the optical flow can only be obtained by subdividing the band into these three regions and estimating the image velocity

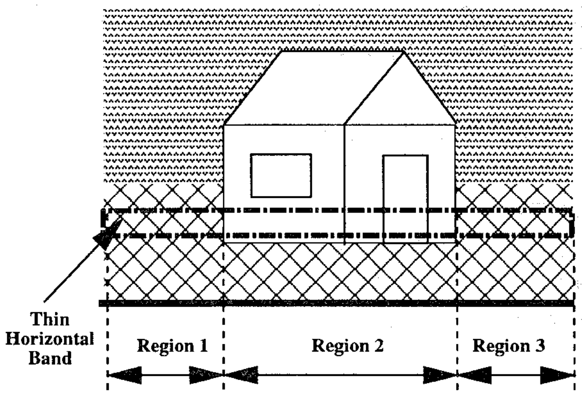

Figure 1: Small Horizontal Band in a Scene containing Motion Boundaries

separately in them by using the whole region and a suitable model[1] for the variation of the flow within it.

In a band, the overall error functional E_{band} is defined as:

$$E_{band} = min\left(\sum_{i=1}^{m} E_R(i) + P_i\right) \tag{1}$$

where the error in each region E_R can be found using a least squares approach to optical flow and a subsequent expansion into a Taylor's series resulting in:

$$E_R(i) = \sum_R W(x,y)(I_t + uI_x + vI_y)^2 \tag{2}$$

The values of u,v that minimize this functional in the region R are the components of the image velocity within a region. This is true as long as the region is homogeneous. The function $W(x, y)$ is an 1d-Gaussian in the vertical direction.

This is a standard formulation, and differentiating Equation (2) with respect to u and v results in

$$\begin{aligned} u\sum WI_x{}^2 + v\sum WI_xI_y &= -\sum WI_xI_t \\ u\sum WI_xI_y + v\sum WI_y{}^2 &= -\sum WI_yI_t \end{aligned}$$

which is a function of the optimal velocities u, v within the regions. The constants P_i are penalty constraints to prevent the functional from resulting in a large number of regions and incorporates a prior belief that the image consists of a small number of objects. Minimizing the functional of Equation (1) is a difficult process as this is non-convex. The solution would have to include finding the optimal number of regions and the location of motion boundaries. Using a translational model for the optical flow distribution within the regions we can solve for the boundary locations with a dynamic programming scheme which leads to a global minimum. This involves solving for the location of one boundary at a time and then traversing the resulting lookup table to find the optimal path. More details can be found in [12].

[1] i.e. translational or affine in most cases

4. APPLICATION OF THIS ALGORITHM TO THE WHOLE IMAGE

The image is split it into horizontal bands and each band is treated separately. To overcome the sensitivity to the actual positioning of the bands we allow for overlapping bands and we use the boundaries calculated before only in the middle of such bands. Typically a band is about 15 pixels high, but the results of the optimization process described are only used to determine the optical flow in the middle 3 to 5 pixels of the band.

The other problem is that the algorithm is anisotropic - the simplest way to handle this would be to obtain results for both vertical and horizontal bands and then to average the two. The difference between the two estimates can be used as a confidence measure. This however would not use the fact that we know where the boundaries are. The method used in this paper is a weighted mean where the weights are equal to the distance from the boundary, the implication being that the flow estimates are more accurate away from boundaries.

5. EXPERIMENTAL RESULTS

The algorithm as described in the previous Section was tested on both synthetic and real image sequences[2]. The penalty function P_i were set to zero - it was found experimentally that if we allowed for enough breaks there was no need to find explicitly how many regions there were. The method of combination of 'vertical' and 'horizontal' flow was modified to test whether a boundary was a real one or just a discontinuity placed by the dynamic programming scheme, in general if the velocity difference either side of this was small it was ignored.

5.1. Synthetic Sequence

The 'square' test sequence was designed to test the algorithm in the presence of well-defined motion boundaries. The sequence consists of four squares translating horizontally with a velocity of 1 pixel/frame and a background which is moving upwards at 1 pixel/frame.

[2] Some of the sequences were obtained by anonymous ftp from Queens' University, Ontario at `csd.uwo.ca`

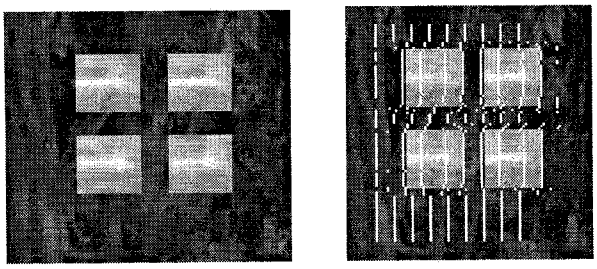

Figure 2: Center frame for the 'squares' image sequence(left), Placement of discontinuities for the horizontal pass(right)

The results shown in Figures 3-5 are for this algorithm and for the output of Lucas & Kanade [3] and Horn & Schunk [2]. Our algorithm used a band size of 5 pixels (with an inner core size of 3 pixels) which is the same effective window size that the Lucas & Kanade algorithm uses.

The discontinuities used by our algorithm are shown in Figure 2; these are within a pixel of the moving square boundary. This is really the limit of the resolution of the algorithm as we are using 5-point derivatives. The optical flow result shows that whereas both the Lucas & Kanade and the Horn & Schunk algorithms smooth right over the motion boundaries, our algorithm does a good job of localizing them. Note especially the sharpness of the transition along the lower horizontal boundary of the lower squares.

5.2. Real Sequences

The image was tested on two sequences, the Hamburg Taxi sequence and the SRI Tree sequence. In the taxi sequence we have four moving objects, a pedestrian in the upper left and the three cars; in the tree sequence and the motion is generated by a camera moving parallel to the scene. It is interesting to note especially in the taxi sequence that the objects are clearly separated and are not enlarged in size - which is generally the case with methods using large windows or global smoothness constraints.

6. CONCLUSION

The overall performance of this algorithm for calculating optical flow is encouraging; it seems to perform very well using roughly the same parameters on a number of very different image sequences. The algorithm is relatively fast and as it is a local method (one band at a time) it would be very easy to design a parallel implementation of it; in the limiting case assign one band to each processor. Future work will include an inves-

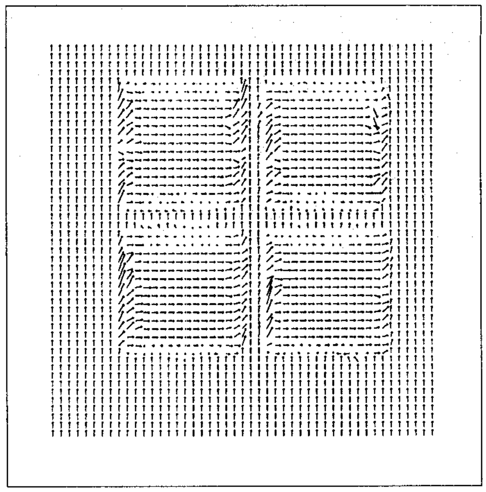

Figure 3: Lucas & Kanade flow for 'squares'

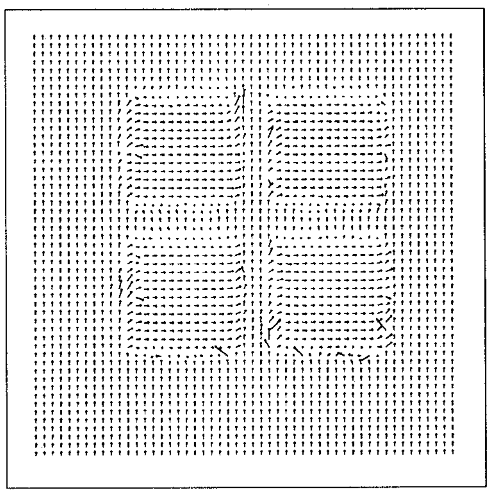

Figure 4: Horn & Schunk flow for 'squares'

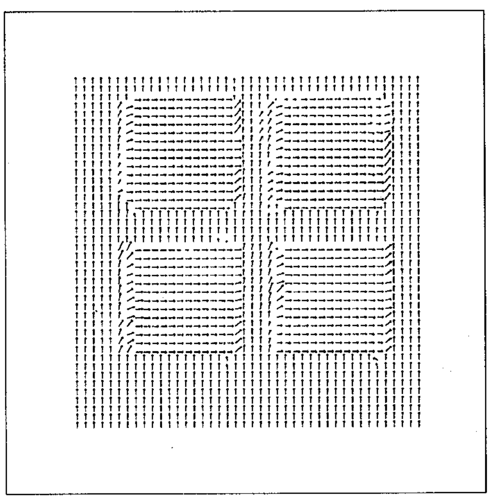

Figure 5: Our algorithm flow for 'squares'

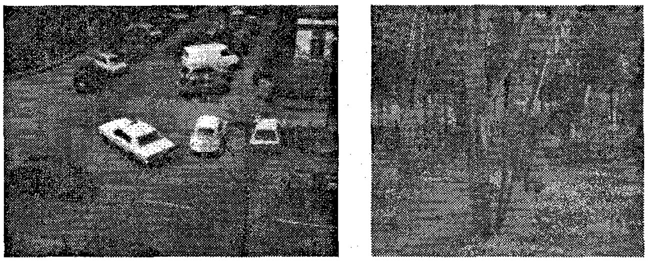

Figure 6: Center frame for the real image sequences

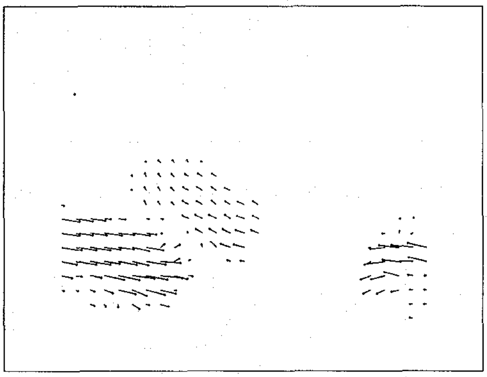

Figure 7: Optical flow for the Hamburg taxi sequence

tigation of how to use edges to guide the discontinuity placement in images without significant texture, such as indoor scenes, where local optical flow estimates become inaccurate.

7. REFERENCES

[1] John Barron, David Fleet, and S.S. Beauchemim. Systems and experiment : Performance of optical flow techniques. *Int. Journal of Computer Vision*, 12:1:43–77, 1994.

[2] B.K.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, vol.17:185–203, 1981.

[3] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of the 7th International Joint Conference on Artificial Intelligence*, pages 674–679, 1981.

[4] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *Int. Journal of Computer Vision*, 2:283–310, 199.

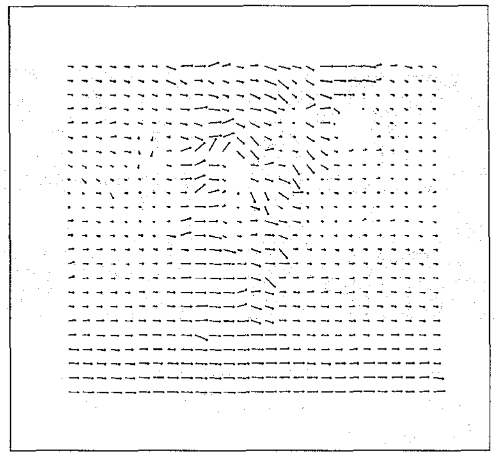

Figure 8: Optical flow for the SRI tree sequence

[5] David J. Heeger. Model for the extraction of image flow. *Journal of the Optical Society of America A*, 4:8:1455–1471, 1987.

[6] Yalin Xiong and Steven A. Shafer. Hypergeometric filters for optical flow and affine matching. In *Proc. Int. Conf. on Computer Vision*, pages 771–776, 1995.

[7] David J. Fleet and Allan D. Jepson. Computation of component image velocity from local phase information. *Int. Journal of Computer Vision*, 5:1:77–104, 1990.

[8] Richard Szeliski and Shum. Heung-Yeung. Motion estimation with quadtree splines. In *Proc. Int. Conf. on Computer Vision*, pages 757–763, 1995.

[9] B.K.P. Horn. *Robot Vision.* McGraw-Hill, NY, 1986.

[10] H.-H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from image sequences. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol.PAMI-8,no.5:565–593, 1986.

[11] C. Schnorr. Computation of discontinuous optical flow by domain decompostion and shape optimisation. *Int. Journal of Computer Vision*, 8:2:153–165, 1992.

[12] X. Papademetris and P. N. Belhumeur. Estimation of motion boundary location and optical flow using dynamic programming. Technical Report 9607, Center for Systems Science,Dept. of Elec. Engineering , Yale University, June 1996.