Fast Computation of a Boundary Preserving Estimate of Optical Flow

Rosario El-Feghali and Amar Mitiche
INRS-TELECOMMUNICATIONS
900, de la Gauchetiere West, Level C
C.P. 644, Montreal, Quebec, H5A-1C6, Canada
{feghali,mitiche}@inrs-telecom.uquebec.ca

Abstract

In this study we seek a fast method for robust, boundary preserving estimation of optical flow. Several studies have addressed this topic and proposed methods that account for velocity boundaries at the cost of significant computational complexity. The answer we bring consists of adapting the benchmark algorithm of Horn and Schunck such that it produces robust boundary preserving estimates, while retaining its simplicity and speed of execution. Experimental verification on real image sequences is conducted.

1 Introduction

The analysis of motion in image sequences can serve a useful purpose in numerous applications. These applications include vision-guided robot autonomous navigation, augmented reality rendering of visual communications data, and visual monitoring of sites of activity. Analysis of motion includes optical flow estimation, a topic that has been the focus of many studies (Mitiche and Bouthemy [15]). Most of the methods proposed for optical flow estimation follow the differential approach, as opposed to correspondence and transforms methods. A differential approach relies on the gradient constraint that relates optical velocity to the image spatiotemporal derivatives. The benchmark of differential methods is that of Horn and Schunck [11]. It is a simple, fast method that produces good estimates except at motion boundaries. It can be implemented to execute in real time (Hutchinson et al [12]). Its only drawback is that it does not preserve motion boundaries. Several studies have addressed the problem of computing boundary preserving estimates. Motion boundaries are accounted for either by the computations underlying the estimation of velocity over the image positional array (Nagel and Enkelmann [16]; Werkhoven and Toet [22]; Nagel [17]; Snyder [20]; Peleg and Rom [19]; Konrad and Dubois [13]; Black [1]; Nesi [18]; Stiller [21]; Black and Anandan [2]; Heitz and Bouthemy [9]; Brailean and A. K. Katsaggelos [3]; Chang et al [4]; Mansouri et al [14]; Hadjres et al [8]), or by prior image brightness segmentation and contour detection (Cornelius and Kanade [5]; Fuh and Maragos [6]; Zheng and Blostein [23]; Gu et al [7]). These approaches can produce better estimates at motion boundaries only at the cost of greater computational complexity, which makes them inadequate for current real-time applications.

In this study we propose to adapt the Horn and Schunck algorithm such that it produces robust boundary preserving estimates, while retaining its simplicity and speed of

BMVC 2000 doi:10.5244/C.14.22

execution. We do so by generalizing the smoothing filter in the Horn and Schunck algorithm so as to vary with position and conform to local variations of velocity.

The remainder of this paper is organized as follows. Section 2 describes the formulation from which alternative algorithms are drawn. Section 3 gives examples of experimental verification, and section 4 is a summary.

2 Formulation

Under the assumption that optical flow is smooth everywhere on the image positional array, the Horn and Schunck method computes an estimate of optical flow that minimizes

$$E = \int\int (I_x u + I_y v + I_t)^2 dxdy + \alpha \int\int (u_x^2 + u_y^2 + v_x^2 + v_y^2) dxdy$$

where (u, v) designates optical velocity, (u_x, u_y) its spatial derivatives, (I_x, I_y, I_t) the spatiotemporal image brightness derivatives, and α is a positive constant that weighs the contribution of the first term of E, which measures the conformity of the estimate to data, relative to the second term, which measures smoothness of the estimate.

The underlying computations on digital images reduce to Jacobi iterations:

$$u_i^{k+1} = \overline{u_i}^k - I_x \frac{I_x \overline{u_i}^k + I_y \overline{v_i}^k + I_t}{\alpha^2 + I_x^2 + I_y^2} \quad (1)$$

$$v_i^{k+1} = \overline{v_i}^k - I_y \frac{I_x \overline{u_i}^k + I_y \overline{v_i}^k + I_t}{\alpha^2 + I_x^2 + I_y^2}$$

where i designates image position, (u_i, v_i) velocity at i, k the iteration index, and $(\overline{u_i}, \overline{v_i})$ the estimate average at i computed according to the mask:

$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$
$\frac{1}{6}$	0	$\frac{1}{6}$
$\frac{1}{12}$	$\frac{1}{6}$	$\frac{1}{12}$

This mask is constant as a result of the underlying assumption that optical flow is smooth everywhere on the image positional array. If it were to vary with position so as to adapt to the local structure of velocity, iterations (1) can compute a robust estimate that accounts for velocity boundaries. In such a case, iterations (1) are executed with:

$$\overline{u}^k = \phi(\{u_j^k\} : j \in \mathcal{N}_i)$$

$$\overline{v}^k = \phi(\{v_j^k\} : j \in \mathcal{N}_i)$$

where ϕ is a smoothing filter adapted to local variations of velocity, and $\mathcal{N}_i$ is the set of neighbors of i (the 4- or 8-neighborhood) . The filter we seek is one that, ideally, will average its arguments over only those points in neighborhood $\mathcal{N}_i$ that are images of points in space that belong to the same object as the point of which i is the image, the assumption being that object surfaces have smoothly varying depth. The problem, therefore, resides in the definition of ϕ. We explore several alternatives that approximate the ideal filter.

2.1 Intensity-based Adaptive Average

Here, $\phi = \phi_I$ produces a weighed average of its arguments, the weight at $j \in \mathcal{N}_i$ being commensurate to the intensity contrast at (i, j).

$$\phi(\{u_j^k\} : j \in \mathcal{N}_i) = \sum_{j \in \mathcal{N}_i} \gamma_j u_j$$

where γ_j is assigned a smaller value for a larger brightness gradient in the direction of i, the central pixel. For instance,

$$\gamma_j = \frac{\frac{1}{1+|I_j - I_i|}}{\sum_{j=1}^{8} \frac{1}{1+|I_j - I_i|}}$$

This is anisotropic diffusion in all digital directions allowed by the form of $\mathcal{N}_i$, eight directions in the case of an 8-neighborhood. The assumption is that the image of environmental objects, or a property of this image, were such a property used instead of the image, has smooth variations, edges appearing only along the projection of objects occlusion boundaries. Under this assumption, velocity edges are identified with image edges. In many cases, of course, velocity edges form only a subset of the image edges. If smoothness is identified minimally with differentiability, diffusion at points interior to a projected surface will be redundant in directions other than those of two orthogonal axes, although this redundancy can but improve the robustness of the estimate to small random variations of the image. At edges, however, anisotropic diffusion in all digital directions will emphasize smoothing of the estimate over all neighboring points on a single side of the edge. Therefore, such diffusion is expected to yield more robust estimates than diffusion in two directions, and simplicity of underlying computations results in fast execution. Under the current driving assumption, ϕ_I, indeed, emphasizes smoothness within regions where intensity variation is smooth and, on the contrary, prevents propagation of velocity estimation across region boundaries.

2.2 Velocity-Based Adaptive Average

Here, $\phi = \phi_W$ produces a weighed average of its arguments, the weight at $j \in \mathcal{N}_i$ being commensurate to the contrast of the current estimates of velocity at i and j.

$$\phi(\{u_j^k\} : j \in \mathcal{N}_i) = \sum_{j \in \mathcal{N}_i} \gamma_j u_j$$

where γ_j is assigned a smaller value for larger velocity gradient in the direction of i, the central pixel. For instance,

$$\gamma_j = \frac{(\frac{1}{1+|u_j - u_i|})^\beta}{\sum_{j=1}^{8} (\frac{1}{1+|u_j - u_i|})^\beta}$$

where $\beta > 1$ is introduced to account for the possibly small range of velocity values. A similar formula is used for v.

The filter is self-referential as it uses weights that are functions of the values of the current estimate of velocity. This is a variational formulation where variations of the estimate are accounted for in all digital directions allowed by the form of the neighborhood. Because variations of velocity of projected surfaces is assumed smooth, and if smoothness is identified minimally with differentiability, accounting for variations in directions other those of two orthogonal axes will be redundant at points interior to projected surfaces. However, at velocity boundaries, ϕ_W will emphasize smoothing over all the points on one side of the boundary. Therefore, ϕ_W is expected to produce a more robust estimate than a variational method that accounts for only two directions, and the simplicity of its underlying computations results in fast execution.

2.3 Median Filtering

Here, $\phi = \phi_M$ is the median of the current estimates in $\mathcal{N}_i$. At a velocity edge, median smoothing is likely to produce a value representative of the values on one side of the edge. An alternative would be to average the values of the estimate in $\mathcal{N}_i$ that are above or below the median, whichever has smaller spread.

2.4 Information Integration

The methods presented can easily integrate information about velocity from other sources by constraining iterations (1) to conform to this information. For instance, they can be constrained to conform to a reliable estimate determined by image feature correspondence at a sparse set of points of the image positional array, or to a reliable estimate computed at image edges by a scheme such as that in Hildreth [10]. The scheme in [10] estimates velocity along zero-crossing contours of the image windowed by a Gaussian. Gaussian smoothing displaces edges, and the method requires that the zero-crossing contours be extracted because velocity estimation refers to each of these contours individually. However, the same reliable estimate can be computed for any edge map by the Horn and Schunck iterations (1) restricted to the set of edges, without need to organize these edges in any way. The computed estimates at edges can then be entered to constrain estimation over the extent of the image positional array. This can be done in several ways. The simplest is to enter the estimate at edges as initial approximation. This estimate at edges can be fixed, not to change with iterations (1), or allowed to change according to iterations (1) executed with ϕ_I, ϕ_W, or ϕ_M. Information integration results in more accurate, boundary preserving estimation of velocity.

3 Experimental Verification

We have run experiments of which we present two examples. One example is that of the sequence Highway, one image of which is shown in Figure 1. The sequence has been acquired with a static camera. The images have little texture, and displacements between successive frames are of the order of several pixels. The other sequence, Bird (Figure 5), is that of a bird on a seashore. The images of this sequence are more textured than those of the Highway sequence, but the texture gradient is low at most places. Both sequences, therefore, are a real challenge to differential methods, and we retained them as examples for this reason. The Bird sequence is a sequence of real images but the movement is

Figure 1: One image of the sequence being tested

synthetically generated (two pixels to the left in a rectangular window around the bird). The Bird sequence is used as a testbed for numerical performance evaluation.

Figures 2 and 3 show the preservation of motion boundaries by ϕ_I, ϕ_W, and ϕ_M for the Highway sequence. The motion boundaries are well delineated as shown in the zoom in at the car and truck. The velocity-based adaptive average ϕ_W has the best performance. We also note that the median method strongly prevents smoothing within the same motion region, and produces sharp velocity boundaries.

Figure 4 shows the results of Horn and Schunck iterations (1) restricted to edges, on a synthetic image sequence which consist of a filled white circle moving diagonally to the lower right of the image by 2 pixels. We find similar results when we compare the results of this algorithm to that of Hildreth [10].

Figure 6 shows the results of combining the velocity-based adaptive average (ϕ_W) with the iterative calculation of velocity at edges. Results obtained are more accurate and preserve velocity boundaries well. Filling in regions of low intensity gradient in moving regions is done better (as shown in the zoom in at the truck Figure 6)

Figure 5 shows that both the velocity-based adaptive average (ϕ_W) and the information integration approach preserve the velocity propagation property within the same motion region. The Table below gives the results for the Bird sequence more objectively. We compute both the average of the horizontal and vertical velocity components (u, v), as well as their respective variances. Iterations using ϕ_I, ϕ_W and ϕ_M execute in approximately the same time, because their difference is confined to the filter ϕ which computes the intensity-weighed average, the velocity-weighed average, and the median of the velocity, in a 3 x 3 neighborhood.

	Horn & Schunck	ϕ_M	ϕ_W	ϕ_I	Inf. Int.
Normalized Mean Error of u	0.133	0.311	0.127	0.144	0.120
Mean Error of v	0.0074	0.0013	0.0057	0.0071	0.0011
Normalized Variance of u	0.0685	0.1243	0.0682	0.0825	0.0410
Normalized Variance of v	0.1659	0.1791	0.1839	0.1985	0.0235

The numerical results concur with the fact that the information integration method with ϕ_W has superior performance.

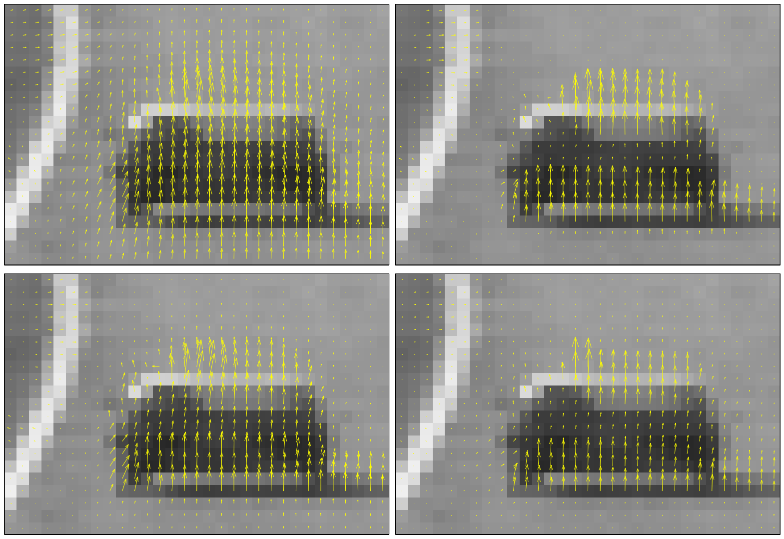

Figure 2: Zoom in at the car : From upper left to lower right, Horn & Schunck, Median-Based Method, Intensity-based Adaptive Average, and Velocity-based Adaptive Average

4 Conclusion

In this paper we have formulated and experimented with different algorithms for fast boundary preserving estimation of optical flow using difficult image sequences. The velocity-based adaptive average has the best performance when compared to the intensity-based adaptive average (ϕ_I), or median smoothing (ϕ_M). Integration of a reliable estimate at image edges improves performance. The methods are fast and can be brought to be implementable to execute in real-time.

References

[1] M. J. Black. Combining intensity and motion for incremental segmentation and tracking over long image sequences. In Giulio Sandini, editor, *Proceedings of Computer Vision (ECCV '92)*, volume 588 of *LNCS*, pages 485–496, Berlin, Germany, mai 1992. Springer.

[2] M. J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Proc. Int. Conf. Computer Vision, ICCV-93, Berlin*, pages 231–236, May 1993.

[3] J. C. Brailean and A. K. Katsaggelos. A recursive nonstationary MAP displacement vector field estimation algorithm. *IEEE Trans. Image Processing*, 4(4):416–429, April 1995.

[4] M. M. Chang, A. M. Tekalp, and M. I. Sezan. Simultaneous motion estimation and segmentation. *IEEE Transactions on Image Processing*, 6(9):1326–1333, 1997.

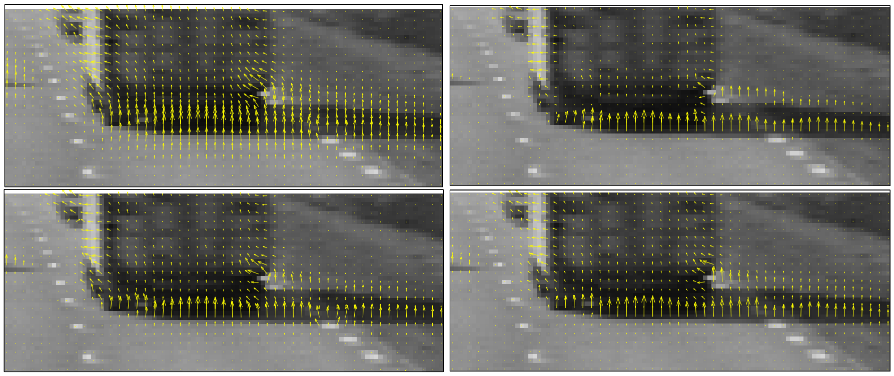

Figure 3: Zoom in at the truck : From upper left to lower right, Horn & Schunck, Median-Based Method, Intensity-based Adaptive Average, and Velocity-based Adaptive Average

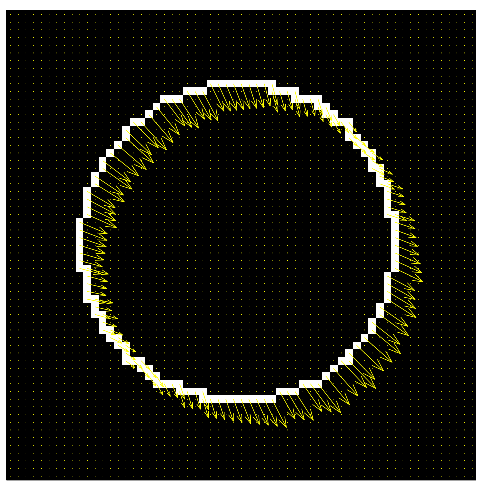

Figure 4: Optical Velocities on Zero-Crossings. The actual motion is one pixel right and one pixel down.

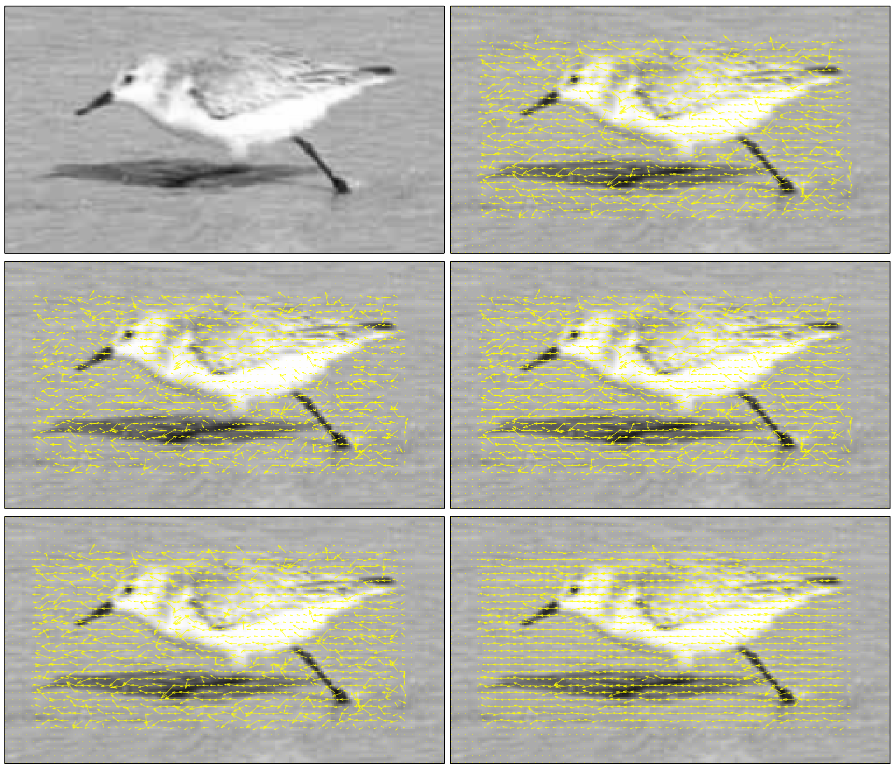

Figure 5: Comparison for intra-region smoothness. From upper left to lower right, the naturally textured bird image, Horn & Schunck optical flow, Median-based Method, Velocity-based Adaptive Average, Intensity-based Adaptive Average, and Information Integration method

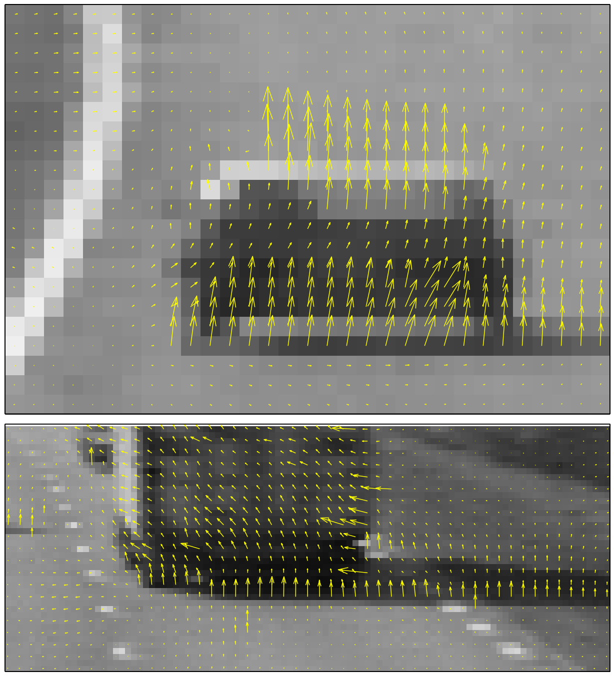

Figure 6: Data Fusion Based Optical Flow

[5] N. Cornelius and T. Kanade. Adapting optical flow to measure object motion in reflectance and x-ray image sequences, 1983.

[6] C. S. Fuh and P. Maragos. Region-based optical flow estimation. In *CVPR'89 (IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, June 4–8, 1989)*, pages 130–135, Washington, DC., June 1989. Computer Society Press.

[7] H. Gu, Y. Shirai, and M. Asada. MDL-based segmentation and motion modeling in a long image sequence of scene with multiple independently moving objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(1):58–64, January 1996.

[8] S. Hadjres, A. Mitiche, and A. Mansouri. Computation of an estimate of optical flow of minimum description length. In *Proc. International Conference on Imaging Science, Systems, and Technology, Session on Statistical and Mathematical Theories in Computer Vision*, pages 411–415, 1999.

[9] F. Heitz and P. Bouthemy. Multimodal motion estimation of discontinuous optical flow using markov random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(12):1217–1232, December 1993.

[10] E. C. Hildreth. Computation underlying the measurement of visual motion. *Artificial Intelligence*, 23(3):309–355, 1984.

[11] B. K.P. Horn and B. G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, August 1981.

[12] J. Hutchinson, C. Koch, J. Luo, and C. Mead. Computing motion using analog and binary resistive networks. *Computer*, 11:52–63, 1988.

[13] J. Konrad and E. Dubois. Bayesian estimation of vector motion fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(9):910–927, 1992.

[14] R. Mansouri, A. Mitiche, and J. Konrad. Selective image diffusion: Application to disparity estimation. In *Proceedings ICIP-98*, pages 284–288, 1998.

[15] A. Mitiche and P. Bouthemy. Computation and analysis of image motion: A synopsis of current problems and methods. *International Journal of Computer Vision*, 19(1):29–55, 1996.

[16] H. Nagel and W. Enkelmann. An investigation of smoothness constraints for the estimation of displacement vector fields from images sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(5):565–593, 1986.

[17] H. H. Nagel. On a constraint equation for the estimation of displacement rates in image sequences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(1):13–30, 1989.

[18] P. Nesi. Variational approach to optical flow estimation managing discontinuities. *Image and Vision Computing*, 11(7):419–439, 1993.

[19] S. Peleg and H. Rom. Motion-based segmentation. In *Proc. 10th International Conference on Pattern Recognition, Atlantic City*, pages 109–113, 1990.

[20] M. A. Snyder. On the mathematical foundations of smoothness constraints for the determination of optical flow and for surface reconstruction. In *Proceedings, Workshop on Visual Motion, (Irvine, CA, March 20–22)*, pages 107–115, 1989.

[21] C. Stiller. A statistical image model for motion estimation. In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, pages 193–196, Mineapolis, 1993.

[22] P. Werkhoven and A. Toet. The estimation of displacement vector fields by means of adaptive affine transformations. In *Proceedings, Eighth International Conference on Pattern Recognition (Paris, France, October 27–31, 1986)*, IEEE Publ. 86CH2342-4, pages 798–800, 1986.

[23] H. Zheng and S. D. Blostein. Motion-based object segmentation and estimation using the MDL principle. *IEEE Transactions on Image Processing*, 4(9):1223–1235, September 1995.