OPTICAL-FLOW BASED ON AN EDGE-AVOIDANCE PROCEDURE

Pierre-Marc Jodoin Max Mignotte

Département d'Informatique et de Recherche Opérationnelle (DIRO), Université de Montréal,

P.O. Box 6128, Succ. Centre-Ville, Montréal, Québec, H3C 3J7.

E-MAIL: JODOINP@IRO.UMONTREAL.CA

ABSTRACT

This paper presents a differential optical flow method which accounts for two typical motion-estimation problems : (1) flow regularization within regions of uniform motion while (2) preserving sharp edges near motion discontinuities i.e., where motion is multimodal by nature. The method proposed is a modified version of the well known Lucas Kanade (LK) algorithm. Based on documented assumptions, our method computes motion with a classical leastsquare fit on a local neighborhood shifted away from where motion is likely to be multimodal. This edge-avoidance procedure is based on the non-parametric mean-shift algorithm which shifts the LK integration window away from local sharp edges. Our method also locally regularizes motion by performing a fusion of local motion estimates. Our method is compared with other edge-preserving methods on image sequences representing different challenges.

Index Terms— Image motion analysis

1. INTRODUCTION

In the past 25 years, a countless number of solutions have been proposed to solve the optical flow problem [1, 2, 3]. As underlined by Barron *et al.* [3], optical flow techniques can be divided into families among which are the phase-based [4], spectral-based [5], energy-based [6, 7, 8], Markovian [9, 10, 11]), and the differential methods [12, 13, 14, 15, 16, 17, 25]. Most of these methods are expressed as an optimization problem involving a *data conservation constraint* likelihood term and a *spatial coherence constraint* prior term [18]. Unfortunately, these competing constraints often ignore the *multimodal* nature of motion around moving edges. As a result, motion is often imprecise and blurry in these areas.

To gain more accuracy around motion discontinuities, a modification to the well known Lucas-Kanade (LK) algorithm [13] is proposed. The objective of our method is twofold : (1) minimize uncertainties (often caused by noise and lack of texture) by strongly constraining the flow within regions of uniform movement while (2) preserving flow discontinuities around moving objects. Since our method is based on a least-square fit (and thus is sensitive to multimodal motion), the key idea is to avoid computing flow in areas where motion is *likely* to be multimodal. Assuming that motion boundaries corresponds to intensity edges, in areas near strong intensity gradient, the algorithm computes motion with a neighborhood window shifted away from the nearest intensity edge. Our least-square fitting algorithm can thus preserve sharp motion boundaries by *avoiding* to deal with multiple motions. To our knowledge, such avoidance procedure has never been investigated before.

The remainder of this paper is organized as follows. Because our method is a modified version of LK, an introduction of the LK method [13] is first presented in Section 2. Section 3 then presents our modifications to LK which includes a covariance filter and the edge-avoidance procedure based on the mean-shift algorithm [19]. Section 4 then presents results obtained on various sequences and concludes.

2. LUCAS-KANADE MOTION ESTIMATION

Let $S = \{s = (i, j) | i \in [0, \mathcal{N}[, j \in [0, \mathcal{M}[\} \text{ denote a 2D lat$ $tice of size } \mathcal{N} \times \mathcal{M} \text{ and } I(s, t) \text{ the intensity of the site } s \text{ at time } t.$ Considering the brightness constancy assumption, LK looks for a vector field $V = \{\vec{v}_s | s \in S\}$ that minimizes the residual quadratic error $E(\vec{v}_s) = \sum_{r \in \eta_s} [I(r, t) - I(r + \vec{v}_s, t + 1)]^2, \forall s \in S \text{ where } \eta_s$ is a neighborhood window of size $N \times N$ around site s. By approximating $E(\vec{v}_s)$ with its first order Taylor serie, the error is minimized by forcing the first derivative to zero : $\frac{\partial E(\vec{v})}{\partial \vec{v}} = 0$ [20], which can be formulated as

$$\left[\begin{pmatrix} \sum_{r} I_x^2 & \sum_{r} I_x I_y \\ \sum_{r} I_x I_y & \sum_{r} I_y^2 \end{pmatrix} \vec{v}_s + \begin{pmatrix} \sum_{r} I_T I_x \\ \sum_{r} I_T I_y \end{pmatrix} \right] = 0$$
(1)

where I_x, I_y and I_T are respectively the spatial and temporal derivatives over site r at time t [21, 20]. It is common to add a weighting factor W_i to give more influence to those site r that a closer to s. This is mathematically expressed as

$$\begin{bmatrix} \begin{pmatrix} \sum_{r} W_{r} I_{x}^{2} & \sum_{r} W_{r} I_{x} I_{y} \\ \sum_{r} W_{r} I_{x} I_{y} & \sum_{r} W_{r} I_{y}^{2} \end{pmatrix} \vec{v}_{s} + \begin{pmatrix} \sum_{r} W_{r} I_{T} I_{x} \\ \sum_{r} W_{r} I_{T} I_{y} \end{pmatrix} \end{bmatrix}$$

where W typically contains Gaussian isotropic values. To simplify the notation, it is common to rewrite this equation as $M_s \vec{v}_s + \vec{b}_s =$ 0 from which the least-square solution can be obtained by simply computing

$$\vec{v}_s = -M_s^{-1}b_s.$$
 (2)

Of course, LK provides a solution to those sites $s \in S$ for which M_s isn't singular. To make M_s invertible everywhere, a small random white noise is added to images I_x and I_y . Since the magnitude of the noise is low, no significant error is induced by it.

To gain more accuracy, some authors implement LK in a Newton-Raphson-like iterative fashion [22]. Let \vec{v}_s^k be the motion vector on site *s* after *k*-1 iterations and $\Delta \vec{v}_s^k$ the incremental motion vector computed during the k^{th} iteration. Here, the goal is to estimate $\Delta \vec{v}_s^k$ that will best minimize the residual error $\sum_{r \in \eta_s} [I(r + \vec{v}_s^k + \Delta \vec{v}_s^k, t) - I(r, t+1)]$. According to Eq. (2), the k^{th} motion increment can be computed by $\Delta \vec{v}_s^k = -M_s^{-1}\vec{b}_s^k$ where

$$\vec{b}_s^k = \left(\begin{array}{c} \sum_r W_r I_T^k I_x \\ \sum_r W_r I_T^k I_y \end{array} \right)$$

$$I_T^k = I(r + \vec{v}_s^k, t) - I(r, t+1).$$

After k iterations, the motion vector on site s is given by $\vec{v}_s^{k+1} = \vec{v}_s^k + \Delta \vec{v}_s^k$ or equivalently $\vec{v}_s^{k+1} = \vec{v}_s^k - M_s^{-1} \vec{b}_s^k$. As explained by Bouguet [22], the iterative version of LK can be easily extended to multiresolution.

3. OUR METHOD

It is widely accepted that LK suffers from two fundamental limitations. First, as shown in the previous section, LK does not model the inherent uncertainties caused by noise and low contrast regions [20]. Also, the LK solution hardly deals with multiple motions and thus generates blurry edges around moving objects.

In this contribution, to minimize the problem of uncertainties, every vector \vec{v}_s are considered as being "estimates" that are to be fused locally to yield a better result [16]. Assuming that \vec{v}_s has a 2×2 covariance matrix C_s proportional to the variance of the noise, the vectors surrounding site s can be fused by a linear combination [8, 16]

$$\vec{v}_s = P_s \sum_{i \in \zeta_s} C_i^{-1} \vec{v}_i \text{ with } P_s = (\sum_{i \in \zeta_s} C_i^{-1})^{-1}$$
 (3)

where ζ_s is a neighborhood window of size $\mathcal{X} \times \mathcal{X}$ around site s and C_s is computed as follows

$$\begin{pmatrix} \frac{\sum_{t \in \beta_s} (u_t - u_s)^2}{\Lambda * \Lambda} & \frac{\sum_{t \in \beta_s} (u_t - u_s) (v_t - v_s)}{\Lambda * \Lambda} \\ \frac{\sum_{t \in \beta_s} (u_t - u_s) (v_t - v_s)}{\Lambda * \Lambda} & \frac{\sum_{t \in \beta_s} (v_t - v_s)^2}{\Lambda * \Lambda} \end{pmatrix}$$
(4)

where β_s is a neighborhood window of size $\Lambda \times \Lambda$ around s and $\vec{v}_t = (u_t, v_t)$. This matrix is similar to the one proposed by Singh [6]. The iterative LK procedure can be rewritten as

$$\vec{v}_s^{k+1} = P_s^k \sum_{i \in \zeta_s} C_i^{k^{-1}} (\vec{v}_i^k + M_i^{-1} \vec{b}_i^k)$$
(5)

which highlights the fact that flow propagates from high-confidence regions (regions with low covariance) to regions of low-confidence.

The way we handle multiple motions is based on the following four assumptions :

- moving objects are textured enough to have their motion correctly estimated by a differential method;
- 2. motion boundaries are close to sharp intensity edges;
- in regions containing no sharp intensity edge, motion is locally uniform;
- motion estimated away from flow discontinuities is reasonably accurate.

From these four assumptions (which are generally accepted in the literature [18, 23]) a fundamental observation can be made : two close sites, that are not separated by an intensity edge, tend to have similar motion vectors. However, flow estimated with a least-square-fit method close to a sharp intensity edge is likely to be corrupted by multiple motions. Thus, with \vec{v}_s , a motion vector estimated by the LK method and \vec{V}_s , the true motion vector on site *s*, it may be inferred that :

- 1. when two neighbors *s* and γ are not separated by an intensity edge and *s* is closer than γ to a motion boundary, then $||\vec{v}_s \vec{V}_s|| \ge ||\vec{v}_\gamma \vec{V}_s||$;
- when the distance between a site γ and the closest motion boundary is larger than N, then v
 _γ ≈ V
 _γ (a similar conclusion was already proposed by Thompson in [23]).

Consequently, to fight against the influence of multiple motions and thus keep sharp edges, Eq. (5) is rewritten as

$$\vec{v}_{s}^{k+1} = P_{\delta}^{k} \sum_{i \in \zeta_{\delta}} C_{i}^{k^{-1}} (\vec{v}_{i}^{k} + M_{i}^{-1} \vec{b}_{i}^{k})$$
(6)

where $\delta = s$ if the distance to the nearest intensity edge is larger than N and $\delta = \gamma$ otherwise. Here, γ is a neighbor of site s located further away from the nearest edges (see Fig. 1). This can be understood as follows : when s is close to an intensity edge, \vec{v}_s estimated with the standard LK approach is likely to be corrupted by multiple motions. It is thus preferable to compute \vec{v}_s with a neighbor window η_{γ} , shifted away from the nearest intensity edges. In this way, \vec{v}_s is computed with a neighborhood that is more likely to contain an unimodal motion.

The question now is, how can γ be estimated ? From the previous four assumptions, it may be derived that a good site γ must respect the following criteria

- 1. γ must be a neighbor of site s with $\mathcal{D}(\gamma) > \mathcal{D}(s)$ (assumptions 2 and 4),
- 2. $||\nabla I(\gamma, t)|| = 0$ (assumption 3),

where $\mathcal{D}(s)$ is the distance between *s* and the nearest intensity edge. From these criteria, we found that the *mean-shift* filtering procedure [19] offers an appropriate strategy to determine γ given *s* and I(t). Mean-shift is a simple iterative nonparametric estimator of density gradient that was first introduced by Fukunaga and Hosteler [24] and adapted to imagery by Comaniciu and Meer [19].

When using mean-shift to filter an image, the iterative procedure is applied on data x_j located in a so-called *spatial-range* domain. The spatial domain refers to the 2D space of lattice S while the range domain refers to the pixel color level. In this case, each site $s \in S$ corresponds to a point x_s in the d-dimensional spatial-range domain (d being 5 for color images and 3 for gray-scale images). After successive mean-shift iterations, the bounding volume initially centered on x_s , is shifted from its initial location x_s to a final position x_{γ} where γ corresponds to a location where the density gradient is null. We call *mean-shift vector*, the vector Γ_s linking site s to site $\gamma : \gamma = s + \Gamma_s$. By the very nature of mean-shift, γ is always located further away from the nearest intensity edge than s and $||\nabla f(x_{\gamma})|| \approx 0$. Also, in general, the stronger the intensity gradient is around site s, the larger Γ_s will be. These are the reasons why we consider that mean-shift meets the two criteria presented previously. To make sure γ is a neighbor of s, Γ_s is clamped to a maximum length : $||\Gamma_s|| = \max(||\Gamma_s||, N)$. Fig. 1 shows a mean-shift vector



Fig. 1. Zoom on a frame of CLAIRE sequence. Every vector shows the estimated mean-shift displacement between a site s and a site γ .

field Γ with vectors linking site *s* to site γ . As can be seen, the closer to an edge a site *s* is, the larger the mean-shift displacement is. For more details on mean-shift, please refer to [19].

4. RESULTS AND CONCLUSION

In this section, results obtained with our method and other differential methods are presented. These results have been obtained on synthetic and real sequences exhibiting clear motion discontinuities. For our method, $\mathcal{X} = \Lambda = N = 9$ and the number of iterations for the Newton-Raphson and the Mean shift procedures is set to four. For every example, the size of the mean-shift bounding volume is set to N in the spatial domain and to 10 in the range domain. Also, our method's weighting factor W has been assigned to $||\nabla I||$ in order to give more influence to regions highly texture. Notice that every method has been implemented in a multiresolution fashion and that all flow fields presented in this section have a density of 100%.

Following Barron *et al.* [3], we implemented the average angular error metric to evaluate the distance between the ground truth vector field \mathcal{V} and the estimated vector field \hat{v} , namely

 $\bar{\psi}_E = \frac{1}{N \times M} \sum_{s \in S} \arccos(\vec{v}_s, \vec{\mathcal{V}}_s)$, where \vec{v}_s and $\vec{\mathcal{V}}_s$ are normalized 3D vectors : $\vec{v}_s = \frac{(u, v, 1)}{\sqrt{u^2 + v^2 + 1}}$. The metric was also implemented on vectors located at a distance lower or equal than 10 pixels of a motion edge, namely $\bar{\psi}_E^e = \frac{1}{N \times M} \sum_{\substack{s \in S \\ \mathcal{D}_s \leq 10}} \arccos(\vec{v}_s, \vec{\mathcal{V}}_s)$, where \mathcal{D}_s is the distance in pixel between site *s* and the nearest motion edge. These metrics are used to evaluate how accurate the optical flow algorithms are near flow discontinuities. The results are also presented in terms of the standard deviation.

As shown in Fig. 2, we have executed our method (as well as five other ones) over two synthetic sequences and two real sequences¹. These sequences were chosen because they exhibit sharp motion discontinuities. The quantitative results are presented in Table 1. We also obtained results on the well known *Yosemite* sequence to illustrate how good our method was to estimate flow on a sequence whose motion isn't in the plane of the camera. These results illustrate visually and quantitatively how precise our method is on such sequences.

Our method shows significant improvement, both around moving edges and on the entire scene. This can be explained by the fact that while mean-shift preserves sharp motion boundaries, the covariance filter smooth out the vector field and thus minimizes errors due to lack of texture, occlusion and noise. Also, by its very nature, our method can be implemented on a parallel architecture such as a programmable graphics card for example. Such implementation is possible because calculation over each site $s \in S$ (at every stage of the algorithm) is independent of its neighbors' processing. The parallel implementation would thus be effective for the mean shift calculation, the motion estimation, and the covariance filtering.

Acknowledgments

The authors are very grateful to Amar Mitiche for its constructive comments, to John Barron for his precious help and to the group of Prof. Dr. H.-H. Nagel for the Taxi sequence.

5. REFERENCES

 Mitiche A. and Bouthemy P., "Computation and analysis of image motion : a synopsis of current problems and methods," *Int. J. Comput. Vision*, vol. 19, no. 1, pp. 29–55, 1996.

¹The C_HS_LK method has been inspired by the method Bruhn *et al.* [25] call the *non linear and multiresolution 2D CLG.*

	With Sky		Without sky	
Technique	ψ_E	σ_{ψ_E}	ψ_E	σ_{ψ_E}
Horn and Schunck[12, 3]	7.35	9.77	3.50	4.44
LK	8.54	14.94	3.42	3.65
Black [18]	5.94	9.31	2.30	1.66
C_HS_LK [25]	7.9	11.20	3.66	3.21
Nagel [14]	8.28	11.55	4.35	5.51
Our method	6.20	13.7	1.81	1.78
Technique	ψ_E	σ_{ψ_E}	ψ^e_E	$\sigma_{\psi^e_E}$
Horn and Schunck[12, 3]	12.0	18.7	33.8	31.2
LK	9.5	20.6	25.9	38.4
Black [18]	8.0	15.7	25.4	28.4
C_HS_LK [25]	9.2	18.3	27.2	34.4
Nagel [14]	10.1	19.2	29.0	35.7
Our method	4.0	16.9	16.2	34.1
Technique	ψ_E	σ_{ψ_E}	ψ^e_E	$\sigma_{\psi_E^e}$
Horn and Schunck[12, 3]	22.1	22.3	36.1	25.2
LK	10.4	21.5	25.4	28.1
Black [18]	14.1	19.4	30.1	23.0
C_HS_LK [25]	8.8	15.8	20.3	20.9
Nagel [14]	22.8	21.9	37.0	25.1
Our method	5.3	15.0	13.2	20.1

Table 1. Results for sequences YOSEMITE, CAR ON PARK and RO-TATING BONSAI.

- [2] Nagel H-H, "Image sequence evaluation : 30 years and still going strong.," in proc. of ICPR, 2000, pp. 1149–1158.
- [3] Barron J., Fleet D., and Beauchemin S., "Performance of optical flow techniques," Int. J. Comput. Vision, vol. 12, no. 1, pp. 43–77, 1994.
- [4] Fleet D. and Jepson A., "Computation of component image velocity from local phase information," Int. J. Comput. Vision, vol. 5, no. 1, 1990.
- [5] Langer M. and Mann R., "Optical snow," Int. J. Comput. Vision, vol. 55, no. 1, pp. 55–71, 2003.
- [6] Singh A., "An estimation-theoretic framework for image-fbw computation," in *Proc. of ICCV*, 1990, pp. 168–177.
- [7] Anandan P., "A computation framework and an algorithm for the measurement of visual motion," *Int. J. Comp. Vis.*, vol. 2, pp. 219–232, 1989.
- [8] A. Singh, "Incremental estimation of image fbw using a kalman filter," J. Vis. Commun. Image Represent., vol. 3, pp. 39–57, 1992.
- [9] Mémin E. and Pérez P., "Hierarchical estimation and segmentation of dense motion fields," *Int. Journal of Computer Vision*, vol. 46, no. 2, pp. 129–155, 2002.
- [10] Heitz F. and Bouthemy P., "Multimodal estimation of discontinuous optical fbw using markov random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 12, pp. 1217–1232, 1993.
- [11] Konrad J. and Dubois E., "Bayesian estimation of motion vector fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 9, pp. 910–927, 1992.
- [12] Horn B. and Schunck B., "Determining optical fbw," AI, vol. 17, pp. 185–203, 1981.
- [13] Lucas B. and Kanade T., "An iterative image registration technique with an application to stereo vision (darpa)," in *Proc. of DARPA Image Unders. Workshop*, 1981, pp. 121–130.
- [14] Nagel H., "On the estimation of optical fbw : relations between different approaches and some new results," *Artif. Intell.*, vol. 33, no. 3, pp. 298–324, 1987.
- [15] Aubert G., Deriche R., and Kornprobst P., "Computing optical flow via variational techniques," SIAM Journal on Applied Mathematics, vol. 60, no. 1, 1999.
- [16] Comaniciu D., "Nonparametric information fusion for motion estimation.," in Proc. of CVPR, 2003, pp. 59–68.
- [17] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck : Combining local and global optic fbw methods," vol. 61, no. 3, pp. 211–231, 2005.
- [18] Black M. and Anandan P., "The robust estimation of multiple motions : parametric and piecewise-smooth fbw fields," *Comput. Vis. Image Underst.*, vol. 63, no. 1, pp. 75–104, 1996.
- [19] Comaniciu D. and Meer P., "Mean shift analysis and applications," in *Proc. of ICCV*, 1999, pp. 1197–1203.
- [20] Simoncelli E., Adelson E., and Heeger D., "Probability distributions of optical fbw," in *Proc. of CVPR*, 1991, pp. 310–315.
- [21] Trucco E. and Verri A., Introductory Techniques for 3-D Computer Vision, Prentice Hall, 1998.



Fig. 2. Results for sequences CARS ON PARK, BONSAI, MOM AND DAUGHTER, TAXI, and CLAIRE. The first two sequences have been computer generated.

- [22] Bouguet J.-Y., "Pyramidal implementation of the lucas kanade feature tracker : Description of the algorithm," Tech. Rep., Intel Corporation, 1999.
- [23] William B. Thompson, "Exploiting discontinuities in optical fbw," Int. J. Comput. Vision, vol. 30, no. 3, pp. 163–173, 1998.
- [24] Fukunaga K. and Hostetler L., "The estimation of the gradient of a density function," *IEEE Trans. on Info. Theory*, vol. 21, pp. 32–40, 1975.
- [25] Bruhn A., Weickert J., and Schnörr, "Lucas/kanade meets horn/schunck : combining local and global optic fbw methods," *Int. J. Comput. Vision*, vol. 61, no. 3, pp. 211–231, 2005.