Refinement of Optical Flow Estimation and Detection of Motion Edges

Andrea Giachetti and Vincent Torre

Dip. Fisica Università di Genova, Via Dodecaneso 33, 16146 Genova
Tel: 39-10-3536311 Fax: 39-10-3536311
E-mail: giach@ge.infm.it

Abstract. In this paper a multi-scale method for the estimation of optical flow and a simple technique for the extraction of *motion edges* from an image sequence are presented. The proposed method is based on a differential neighborhood-sampling technique combined with a multi-scale approach and flow filtering techniques. The multi-scale approach is introduced to overcome the aliasing problem in the computation of spatial and temporal derivatives. The flow filtering is useful near motion boundaries to preserve discontinuities. A *residual function*, which is a confidence measure of the least-squares fit used to compute the optical flow, is introduced and used to filter the flow and to detect motion boundaries. These boundaries, that we call *motion edges* are extracted by searching for the directional maxima of the map obtained by thinning this residual function. The proposed method has been tested in a variety of conditions. The results obtained with test images show that the proposed approach is an improvement of previous techniques available in the literature.

Introduction

The computation of the optical flow with a precise detection of image locations where the motion field presents discontinuities is a relevant and hard problem of Computer Vision. Because the gray level evolution at one pixel is not sufficient to determine the flow at its location, all the methods used to compute the image motion must use information coming from the surrounding pixels, making assumptions of constancy or smoothness of the motion in these points. *Neighborhood-sampling* optical flow estimators [2, 3, 5] provide a simple way to roughly identify the image locations where motion discontinuities occur. These methods assume that the flow is constant or linear over a small window centered in the considered point, writing for each pixel of the window an equation relating the velocity components to the image derivatives (usually the gray-level constancy constraint $dE/dt = 0$ [1]) and solving the obtained overconstrained system with standard least-square techniques. If the flow is discontinuous in the window, a low confidence measure of the least-square fit is found and tests on the quality of the fit can be used for the detection of motion discontinuities. Nagel [6], suggests the use of stochastic tests on parameters measured by a flow estimator to detect the motion boundaries. A confidence measure of the fit can also be used to improve the accuracy of the computed flow near the discontinuities as proposed by Bartolini et al. [8]. However, to detect motion discontinuities and improve the accuracy of the flow, other sources of error should not affect the confidence measure. The flow can be wrong for at least other three reasons: if the brightness constancy equation is not satisfied, i.e. $dE/dt \neq 0$; if the approximation of the derivatives with finite differences is not good; if the texture in the window is poor or ambiguous (aperture problem). The analysis of these errors suggested us to develop a refined optical flow algorithm based on a multiscale approach [11, 9, 10] combined with edge-preserving flow filters [8]. We

also realized a new algorithm for the localization of "motion edges" based on a search for the local directional maxima of a refined *residual function* which is a local measure of the error of the multi-scale estimate. When the shape of the motion boundaries is available because of some a priori information, a better detection of motion boundaries can be obtained by a parametric estimation of shape templates through the detected motion edges.

1 The basic optical flow algorithm

Fitting the derivatives to the constant local velocity model - Let us assume that that the condition $dE/dt = E_x u + E_y v + E_t = 0$ linking the optical flow components $\mathbf{v} = (u, v)$ to the derivatives of the grey level E: E_x, E_y, E_t holds true over a window centered in the considered point. In detail, given the spatio-temporal window W with $n \times n \times m$ pixels centered in the point we have:

$$\tilde{E}_x(i,j,k)u(x,y,t) + \tilde{E}_y(i,j,k)v(x,y,t) = -\tilde{E}_t(i,j,k) \qquad (i,j,k \in W) \quad (1)$$

where $\tilde{E}_x, \tilde{E}_y, \tilde{E}_t$ are the estimated derivatives. Applying the standard least-squares technique we obtain as the best estimate $\tilde{\mathbf{v}}$ the one minimizing the quantity: $\sum_{i,j,k \in W}(\tilde{E}_t(i,j,k) + \tilde{E}_x(i,j,k)u(x,y,t) + \tilde{E}_y(i,j,k)v(x,y,t))^2$,

that is: $\tilde{\mathbf{v}} = (\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{Y}$ with $\mathbf{B} = \begin{pmatrix} \tilde{E}_x(\mathbf{x}_1) & \tilde{E}_y(\mathbf{x}_1) \\ \dots & \dots \\ \tilde{E}_x(\mathbf{x}_n) & \tilde{E}_y(\mathbf{x}_n) \end{pmatrix}$.

$\mathbf{B}$ is the "coefficient matrix", where $(\mathbf{x}_1 \dots \mathbf{x}_n)$ are the coordinates of the pixels inside the window, while $\mathbf{Y} = (-\tilde{E}_t(x_1), \dots, -\tilde{E}_t(x_n))$. is the "data" vector. Supposing that eqn. (1) holds true, the estimated optical flow is close to the true image velocity only if: 1) the system is well conditioned; 2) the assumption of constant $\mathbf{v}$ holds true. Condition 2 suggests the use of small windows, so as to reduce the number of wrong estimates due to the presence of discontinuities inside the window. In this case, however, the equations of the system are heavily correlated due to the effect of the derivative mask and of image smoothing[10]. In order to have a well conditioned system larger windows must be used; to reduce the computational weight, as the data from neighboring points are strongly correlated, it is possible to eliminate some equations by subsampling the windows as suggested in Fig. 1. Large windows are not sufficient to guarantee a correct estimate. In fact, in order to compute a reliable flow, spatial derivatives must be nonzero in more than one point and the ratio E_x/E_y should not be constant. This can be ensured by requiring a nonzero determinant of the matrix $\mathbf{B}^T\mathbf{B}$. To eliminate wrong estimates it is necessary to have a threshold d_T on the value of $Det\ (\mathbf{B}^T\mathbf{B})$. A threshold on the conditioning number c_T, defined as the absolute value of the ratio between the minimum and the maximum eigenvalue of the matrix avoids numerical instability of the inversion of $(\mathbf{B}^T\mathbf{B})$ [7].

Accuracy of the fit - The quality of the fit can be evaluated by analyzing the value of the *residual function*:

$$Q(W(\mathbf{x})) = \sum_{i,j,k \in W} (\tilde{E}_t(i,j,k) + \tilde{E}_x(i,j,k)\tilde{u}(x,y,t) + \tilde{E}_y(i,j,k)\tilde{v}(x,y,t))^2/N(W) \quad (2)$$

where N(W) is the number of the pixel of the window used for the estimation. High values of this function indicate a bad fit of the model and an inaccurate flow field. This is the case when there are points in the window where the gray level constancy is false or where the estimation of the derivatives is wrong or if there is a sharp motion discontinuity inside the window. In the following we will always

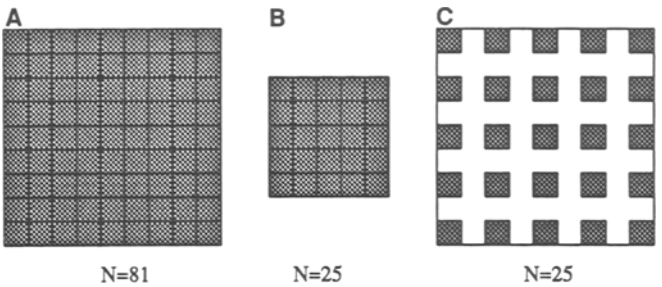

Fig. 1: Advantages of using sub-sampled windows. A: $9 \times 9(\times 1)$ window: the size is almost always sufficient to have a well conditioned system, but the number of equation is high. B: 5×5 window: the computation is speeded up, but the conditioning of the system is often poor. C: 9×9 subsampled window: the number of equations is the same as in B, but the system is almost always well conditioned.

consider condition (1) to hold. If this is true and the estimates of the derivatives are correct, Q can be then used for the location of motion discontinuities. In order to do this it is necessary to analyze carefully the problems in computing derivative estimation.

2 Errors in computing derivatives

Two main kinds of errors affect derivatives computed from finite difference or polynomial fitting: errors due to poor texture, and errors due to the spatio-temporal sampling steps.

Poor texture - It is evident that, whenever a spatial derivative is almost zero, eqn. (1) becomes meaningless because of the large relative errors. Let us consider, the three point approximation of the derivatives: $\tilde{E}_x(x,y) = [E(x+\delta x, y) - E(x-\delta x, y)]/2\delta x$ and suppose that the first order approximation of the signal is good so that the use of the finite difference does not introduce error. The only error will be due to the uncertainty on the values of E. Let us suppose that this uncertainty is limited by the quantity δE; then the error on the derivative will be $\delta E/\delta x$. The relative error is then:

$$\delta E_x / E_x = \delta E / (\tilde{E}_x \delta x) \tag{3}$$

that is high when local variations of E are negligible. A similar analysis show that the second order derivatives computed with the standard 5-points mask are much more sensitive to noise: in fact the relative error is $16\delta E/3\tilde{E}_{xx}\delta^2 x$ and the average value of E_{xx} is usually lower than the average value of E_x.

Spatio-temporal sampling / Aliasing - The recovery of the derivatives from a sampled signal is not always possible, because the knowledge about the original signal is not complete due to sampling.
To see the effect of high frequencies on the derivatives of our sampled image we can compare the estimated derivatives in the point $(0,0,0)$ (we choose this point to simplify the formulas, but it would be the same for a general location) with their true values. We consider a neighborhood I of $(0,0,0)$ defined by $|x-ut| < \lambda, |y-vt| < \lambda, |t| < \tau$, so that, assuming $\mathbf{v}$ constant in I, the signal can be written as [9]: $E(x,y,t) = f(x-ut, y-vt)$, where f can be expressed by the Fourier series:

$$f(x-ut, y-vt) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} A_{nm} \exp(j(n\omega(x-ut) + m\omega(y-vt))) \tag{4}$$

where $\omega = 2\pi/\lambda$ is the fundamental frequency of the signal and the coefficients A_{nm} are defined by integrals of the function multiplied by exponentials as usual. The derivative along the $x-$ direction in $0,0,0$ is then:

$$E_x = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} A_{nm} jn\omega \tag{5}$$

The three point approximation of the derivative is:

$$\tilde{E}_x|_{(0,0,0)} = \frac{f(\delta x, 0) - f(-\delta x, 0)}{2} =$$
$$= \sum_{n,m=-\infty}^{\infty} A_{nm} \frac{e^{jn\omega\delta x} - e^{-jn\omega\delta x}}{2\delta x} = \sum_{n,m=-\infty}^{\infty} A_{nm} j \frac{\sin(n\omega\delta x)}{\delta x} \tag{6}$$

The difference between the estimated and the true value is:

$$E_x - \tilde{E}_x = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} A_{nm} j(n\omega\delta x - \sin(n\omega\delta x)) \tag{7}$$

We immediately notice that the estimated derivative is correct only if all the coefficients A_{nm} corresponding to frequencies $n\omega << 1/\delta x$ are negligible. The approximation of the derivative is then good only if for the maximum frequency with relevant energy, $N\omega$, we have: $\sin(N\omega\delta x) \simeq N\omega\delta x \quad i.e. \quad N\omega\delta x << 1$
When many frequencies over this limit present relevant coefficients, the relative error may be in the order of the unity. The analysis of the spatial derivative suggests to remove frequencies higher than $1/\delta x$ to avoid large errors. With the same method used before, it is possible to see that difference between the true value and the 3-point estimate of the temporal derivative in (0,0,0) is:

$$E_t - \tilde{E}_t = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} A_{nm} j \frac{n\omega u\delta t + m\omega v\delta t - \sin(n\omega u\delta t + m\omega v\delta t)}{\delta t} \tag{8}$$

The estimate is correct only if for the highest frequencies $N\omega, M\omega$ with relevant coefficients A_{NM}, we have: $\sin(N\omega u\delta t + M\omega v\delta t) \simeq \omega(Nu + Mv)\delta t$
To eliminate all the frequencies introducing large errors in the estimates, it is therefore necessary to use a low-pass filter with a cutoff frequency chosen according to the largest value of **v** to be detected.

3 Multi-scale techniques

A good solution to the problem of aliasing without the loss of all the high-frequencies information is the *multi-scale* approach [9, 11]. Multi-scale techniques are usually based on the "pyramid" decomposition of the images. Each of the original $l_x \times l_y$ images is smoothed and then subsampled keeping only even or odd pixels. The new $l_x/2 \times l_y/2$ images can then be decomposed in the same way until the coarsest resolution $l_x/2^N \times l_y/2^N$ is obtained. Moving from the finer $(l_x \times l_y)$ to the coarser $(l_x/2 \times l_y/2)$ image removes higher frequencies. To avoid the poor estimate due to the derivative errors, it is necessary to stop the decomposition at a resolution where the motion is in the order of 1 pixel/frame. The coarsest resolution, however, should be sufficiently fine to keep the assumption of flow constancy in the windows well approximated. We always used two or three resolutions in the experiments, enough to compute the largest motions

without introducing large errors. In the multi-scale flow computation, the integration of the information relative to the different scales is performed in a *coarse to fine* framework. In [9] the flow is computed at finer scales only where the one computed at the coarse scale is considered wrong. We have chosen a different solution, consisting of computing the flow at the coarser resolution and then compute at the finer resolution corrective terms, i.e. differences between the flow and the integer approximation of the previously computed flow, with a scheme recently introduced by Xu [11]. Let us denote with $\mathbf{V}^N = (U^N, V^N)$ the integer approximation of the flow $\mathbf{v}^N$ computed at the resolution $l_x/2^N \times l_y/2^N$. If at the finer resolution $l_x/2^{N-1} \times l_y/2^{N-1}$, we compute the *shifted derivatives* defined as follow:

$$E_t'^{N-1} = \frac{E(x + U_x^N, y + V_x^N, t+1) - E(x - U_x^N, y - V_x^N, t-1)}{2} \tag{9}$$

in all the points of a window $W(\mathbf{x})$ centered in the point $\mathbf{x}$ of interest; from the overconstrained system

$$E_x^{N-1}(\mathbf{x})c_x + E_y^{N-1}(\mathbf{x})c_y + E_t'^{N-1}(\mathbf{x}) = 0 \qquad (\mathbf{x} \in W(\mathbf{x})) \tag{10}$$

it is possible to compute the *corrective terms* c_x, c_y, representing the additional term to be added to the integer values U^N, V^N. If the residual function $Q^{N-1}(\mathbf{x})$ of this fit is less than the residual fit $Q^N(\mathbf{x})$ computed estimating the value of $\mathbf{v}^N$, we consider as the best estimate of the flow in $\mathbf{x}$ at the scale $N-1$, the quantity described by:

$$u^{N-1} = U^N + c_x^{N-1} \qquad v^{N-1} = V^N + c_y^{N-1} \tag{11}$$

and we replace the value of the residual Q stored in memory with the new value Q^{N-1}. In this way information coming from finer scales is used without introducing further error. The coarse to fine velocity propagation is not correct if a a pixel of the coarser scale correspond to a motion discontinuity. Near these points it is necessary to improve the quality of the estimation before computing the corrective terms, and use some special procedure for the flow correction (see Section 5).

4 Discontinuities

Detection of discontinuities - When a discontinuity occurs in the window (or close to a window) used for the estimation of $\mathbf{v}(\mathbf{x})$, the result obtained is wrong for two reasons: firstly, the assumption of constant $\mathbf{v}$ inside the window fails; secondly, the derivative estimates obtained where the discontinuity is located inside a derivative mask are not reliable. Supposing to have eliminated the effects of the errors in derivative computation described before, we can identify the regions where a sharp motion discontinuity occurs, with the pixels where the the residual function $Q(W(\mathbf{X}))$ presents high values. $Q(W(\mathbf{X}))$ is, in this case, approximately zero when the whole window is inside a coherently moving region, nonzero when a discontinuity crosses the window and presents its directional maxima where a discontinuity passes near the center of the window. We can roughly identify these maxima with the "motion edges".

Flow filtering near discontinuities: the multi-window approach - The residual function can also be used to improve the flow accuracy near the discontinuities, by using a non linear filter. A similar approach has been successfully introduced by Bartolini et al. [8]. Their algorithm is very simple: for each pixel

location $\mathbf{x}$ 9 windows containing the point and shifted as shown in Fig. 2 A are considered. The estimates obtained in the points $x - mask * i, y - mask * j$ with $i, j = -1, 0, 1$ and the corresponding average squared residuals are considered ($mask$ is equal to half of the mask dimension). The correct flow estimated in $\mathbf{x}$ is considered to be the one obtained in the window with the lowest residual function. In this way a window containing the central point and with uniform motion inside is searched in the eight directions and the quality of the estimates is improved.

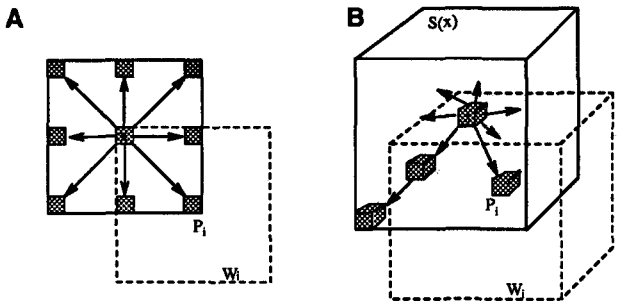

Fig. 2: A: The multi-window filter of Bartolini et al. It is assumed as velocity estimate in the central point the one obtained in the window centered in one of the nine points where the residual function is lower. B: Generalization of the residual filter: the minimum of the residual function is searched in a 3D (x,y,t) search space S including pixels where the flow estimate depends on the gray value in the central point. The best velocity estimate is the one obtained where the residual function is minimum.

The generalized residual filter - This filtering procedure can be generalized with a search for the minimum value of the residual on a set of points of the neighborhood of $\mathbf{x}$ corresponding to masks containing $\mathbf{x}$ or where the velocity estimate depends on $E(\mathbf{x})$ because of the extension of the derivative mask. It is important to extend the search for continuity for our filter even in the temporal direction, to compensate for the effects of temporal discontinuities of the motion. In our filter the search for the minimum of the residual function is performed in a search space $S(\mathbf{x})$ of $9 \times 9 \times 3$ pixels (the mask dimension used is $9 \times 9 \times 1$). The filtered flow estimate $\mathbf{v}_{rf}(\mathbf{x})$ in the point $\mathbf{x}$ is therefore chosen as the one obtained in the point of $S(\mathbf{x})$ where the residual function is minimum.

Regularization preserving the edges - We introduced also a flow regularization preserving the sharp variations of the flow near the discontinuities. First the average $\mathbf{v}_{avg}$ of the flow vectors over a set of points of the neighborhood of $\mathbf{x}$ is computed. This set is composed by pixels $\mathbf{x}'$ where the residual function is smaller than a threshold and the difference between the computed flow $\mathbf{v}(\mathbf{x}')$ and the output of the residual filter $\mathbf{v}_{rf}(\mathbf{x})$ is less than 1 $pixel/frame$. The regularized estimate of $\mathbf{v}_{reg}(x)$ is given by: $\mathbf{v}_{reg}(x) = (\mathbf{v}_{rf} + \mathbf{v}_{avg})/2$

Thinning of the residual map and edge detection - The final multi-scale residual map $Q(\mathbf{x})$ identifies the presence of discontinuities, but its localization is poor because of the window size. It is possible, however, to obtain a thinning of this map during the residual filtering, by assuming as new residual value $Q'(\mathbf{x})$ at each pixel location the minimum value of the residual in the search space $S(\mathbf{x})$. Motion discontinuities are still revealed because the size of the region around $\mathbf{x}$ where the flow estimate depends on $E(\mathbf{x})$ is larger than the size of the search space. Our algorithm for the detection of *motion edges* is therefore based on the search for local maxima along $x-$ and $y-$ directions of the map $Q'(\mathbf{x})$, larger than a defined threshold.

5 Description of the complete multi scale algorithm

The above analysis suggested us to design a multi-scale algorithm for the detection of motion edge and flow estimate, consisting of the following steps:

1. The image sequence is filtered and decomposed in a Gaussian pyramid.
2. The optical flow is computed at the coarsest scale over $9 \times 9 \times 1$ (subsampled) windows, rejecting equations obtained where spatial derivatives (computed with a 5 point mask) are below a threshold and rejecting estimates obtained where the determinant and the conditioning number of the matrix $\mathbf{B}^T\mathbf{B}$ are less than the thresholds d_T and c_T and the residual function is above the threshold r_T.
3. The residual filter is applied to the flow.
4. The regularizing filter is applied to the flow;
5. Flow corrections are computed at the finer scale. If the residual function computed at the coarser scale had a high value, several corrective terms are computed shifting the derivatives not only of $(int)v(x, y, t)$, but also of $(int)v(x + i, y + j, t + k)$ with $i, j, k = -1, 0, 1$. The shift corresponding to the minimum value of the residual function at the finer scale is kept.
6. The corrected flow at the finer scale is kept if its residual is lower than the one computed at the coarser scale, then the residual map Q is corrected with the new value. If the residual is higher, the flow computed at the coarser scale is still considered the best estimate.
7. Filtering is applied to the corrected flow and the procedure is repeated until the finer scale is reached.
8. When the highest resolution is reached, the thinned residual map is generated as described in Section 4 and the motion edges are extracted from this map.

6 Experimental results

We have tested our algorithms on calibrated sequences (with known true displacements). To evaluate the accuracy of the computed flows we used the angular distance between the computed $\mathbf{v} = (u, v)$ and the real $\mathbf{v}' = (u', v')$ displacements, defined by Barron [4] as: $dist(\mathbf{v}, \mathbf{v}') = \arccos\left(\frac{uu'+vv'+1}{\sqrt{(|\mathbf{v}|+1)(|\mathbf{v}'|+1)}}\right)$

Enhancement of flow accuracy - The first series of experiments aimed at testing the proposed optical flow algorithm. We have measured the improvement in the flow accuracy on a synthetic sequence where a circle with radius equal to 100 pixels translates with velocity $(1, 0.5)$, over a background translating with velocity $(0, -0.5)$, and on the Marbled Block sequence created by Michael Otte (Karlsruhe), a calibrated sequence where the camera is moved while the white block is translating (a reduced version of $256 \times 256 pixels$). All the parameters are known, so the correct map of the image displacements is known. In the first case, using the single scale algorithm with 9×9 windows and gaussian smoothing with $\sigma = 1.5$, the average angular distance was 4.59^o degrees (100% density). With the two scale algorithm (N=2) the distance was reduced to 3.97^o. Introducing the residual filter at each scale the distance became 3.52^o and adding the regularizing filter 2.92^o. For the marbled block sequences the average angular distance from the true displacements was 7.77^o (100% density) with the basic estimator, 7.45^o introducing the two scale decomposition, 6.65^o and 5.33^o with the residual filter and both the residual and regularizing filters. Another test has been done to demonstrate the possibility of speeding up the computation by window subsampling. Table 1 shows the average angular differences from the true motion of flows computed on the synthetic sequence of Fig. 4 in different cases. Reducing the number of equations by subsampling does not lead to a deterioration of the flow.

Window	dist	std.dev.	dens.
9 × 9 (81 eqns)	2.63°	6.71°	82.6
5 × 5 (25 eqns)	4.33°	7.49°	69.2
9 × 9 sub. (25 eqns)	2.86°	7.32°	77.1
9 × 9 sub. (9 eqns)	4.23°	8.33°	65.3

Table 1: If the number of equations is reduced by reducing the window size, the estimate gets poorer, while subsampling a window of the same size, the equations are reduced with no relevant effects on flow accuracy.

Quantitative comparison with other techniques - We have compared the average angular differences between our flows and the corresponding true displacements on some test sequences. The Translating Tree and Diverging Tree sequences, created by David Fleet, are synthetic sequences simulating the translation perpendicular and parallel to the optical axis of the camera of a slanted planar surface. In these sequences no motion discontinuities are present, but the presence of large motions (~ 2 pixel/frame) shows the utility of the multi-scale approach: the accuracy of our results is satisfactory compared with the values reported in[4, 10, 11] (Table 2). The "Yosemite Valley" sequence (Fig. 3 A), created by Lynn Quam, is a challenging test for optical flow algorithms, because it presents large displacements, occlusions and regions with poor texture. The flow obtained with our algorithm (Fig. 3 B) is more accurate than those obtained with other techniques (Table 2).

Algorithm	Translating Tree			Diverging Tree			Yosemite		
	ang.d.	std.dev.	dens.	ang.d.	std. dev.	dens.	ang.d.	std.dev.	dens.
Horn & Schunck	38.72°	27.67°	100	12.02°	11.72°	100	32.43°	30.28°	100
Heeger	4.53°	2.41°	57.8	4.49°	3.10°	74.2	10.51°	12.11°	15.2
Anandan	4.54°	2.98°	100	7.64°	4.96°	100	15.84°	13.46°	100
Lucas & Kanade	0.66°	0.67°	39.8	1.94°	2.06°	48.2	4.10°	9.58°	35.1
Fleet & Jepson	0.32°	0.38°	74.5	0.99°	0.78°	61.0	4.25°	11.34°	34.1
Weber & Malik	0.49°	0.35°	96.8	3.18°	2.50°	88.6	4.31°	8.66°	64.2
Xu	1.89°	4.23°	95.7	4.11°	6.56°	93.7	9.93°	11.03°	99.8
Our Impl.(raw)	0.51°	0.46°	95.0	3.97°	2.60°	95.0	4.01°	7.12°	70.9
Our Impl.(reg.)	0.25°	0.23°	95.0	2.07°	1.37°	95.0	2.82°	6.98°	70.9

Table 2: Quantitative comparison of the flow accuracy obtained on the sequences Translating Tree, Divering Tree and Yosemite with our algorithm and with other techniques.

Detection of motion edges - The final series of experiments aimed at detecting motion edges through the search for the directional maxima of the thinned residual function. Fig. 4 A shows the residual function obtained from the computation of the single - scale flow on the synthetic sequence described before. High values of the residual function do not correspond only to motion discontinuities due to the error in derivatives computation. With the multi-scale approach, the residual map clearly presents its highest values near the discontinuities (Fig. 4 B). Applying the residual function thinning as described in Section 5, a clear localization of the boundaries of the moving region is found (Fig. 4 C). The "motion edges" extracted from this map are finally superimposed to the original image in Fig. 4 D showing their proximity to the true boundaries of the moving region. In the case of the *Marbled Block* sequence, it is much more difficult to estimate the position of the motion boundaries, because flow differences between different objects are not so relevant. The refined residual map obtained from the reduced 256 × 256 sequence with our multi-scale algorithm is shown in Fig. 4E. In spite of the difficulties, the boundaries of the nearest column and of the marbled block are clearly revealed (Fig. 4 F).

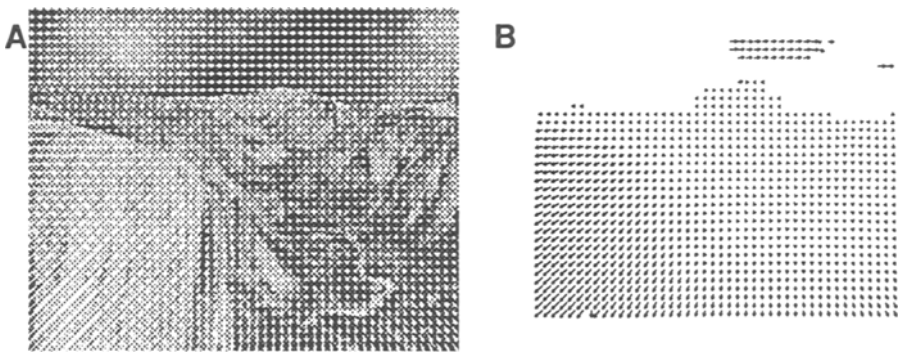

Fig. 3: Results obtained on the Yosemite Valley sequence. A: Sequence image with the true motion superimposed. B: Optical flow computed with our algorithm.

Use of shape models - The *motion edges* extracted with our technique are close to the boundaries of the moving regions, but are inevitably irregular and incomplete due to the texture properties and to errors. In order to refine the detection of these boundaries it is therefore necessary to introduce other information and other processing. If templates of the moving objects are known, a possible solution is a parametric fitting of the model through the detected edges (Fig. 5 A,B).

7 Conclusions

Our work shows that motion edges can be extracted and the quality of the optical flow estimation can be improved without the introduction of complex techniques. In a similar way also motion discontinuities can be satisfactorily revealed.

References

1. B.K.P.Horn & B.G.Schunck, "Determining optical flow," Art. Int. **17**, 185–203 (1981).
2. B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision" Proc. DARPA Image Und. Workshop 121–130 (1981).
3. M. Campani and A. Verri, "Motion analysis from first order properties of optical flow," CVGIP - Image Understanding **56**, 1:90–107 (1992).
4. J.L. Barron, D.J. Fleet and S.S. Beauchemin, "Performance of optical flow techniques" Int. J. Comp. Vision **12**, 1:43–77 (1994).
5. M. Otte and H.H. Nagel, "Optical flow estimation: advances and comparisons" Proc. Third ECCV, **14**, 51–60 (1994).
6. H. H. Nagel, "Optical Flow Estimation and the Interaction Between Measurement Errors at Adjacent Pixel Positions" Int. J. Comp. Vision **15**:3, 271–288 (1995).
7. E.De Micheli, S.Uras & V.Torre, "The accuracy of the computation of optical flow and of the recovery of motion parameters," IEEE Trans. PAMI **15**, 5:434–447 (1993).
8. F.Bartolini, V.Cappellini, C.Colombo, A.Mecocci, "Multiwindow least-square approach to the estimation of optical flow with discontinuities" Opt. Eng., **32**, 6: 1250–1256, (1993).
9. R. Battiti, E. Amaldi and C. Koch, "Computing Optical Flow Across Multiple Scales: An Adaptive Coarse to Fine Strategy", Int. J. Comp. Vision **6**:2, 133–145 (1991).
10. J. Weber and J. Malik "Robust Computation of Optical Flow in a Multi-Scale Differential Framework" Int. J. Comp. Vision **14**, 1:67–81 (1995).
11. S. Xu "Motion and Optical Flow in Robot Vision" Linköping Studies in Science and Technology - Thesis No. 442 (1994).

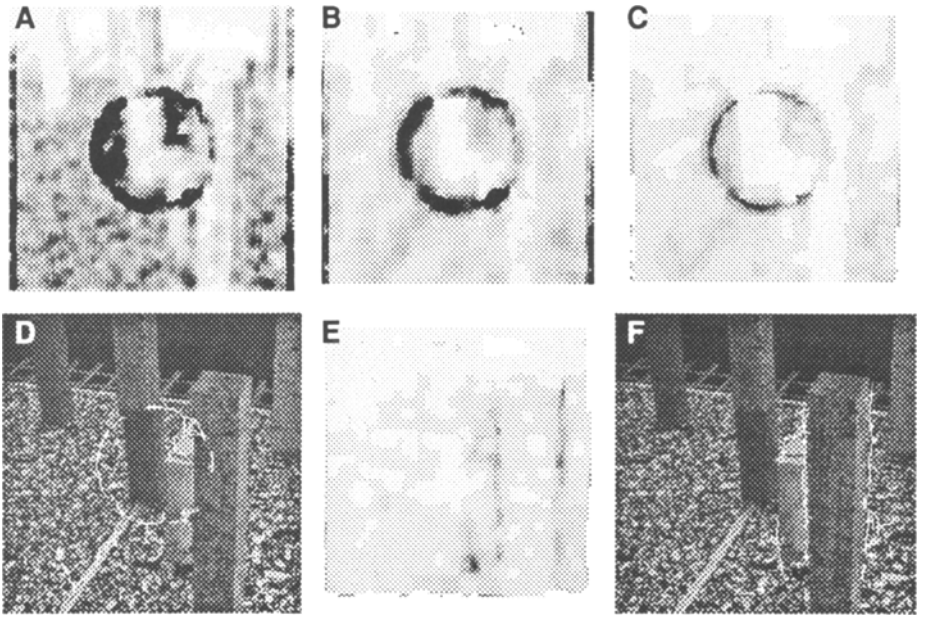

Fig. 4: Extraction of "motion edges". A: Residual function obtained from the single-scale flow computation on a synthetic with the marbled block texture. A central circle and the background are differently translating. B: Residual function obtained from the multi-scale algorithm on the same sequence. C: "Thinned" residual map. D: "Motion edges" superimposed to the image. E: Thinned multi-scale residual map obtained on the 256×256 Marbled Block sequence. F: "Motion edges" superimposed to the image.

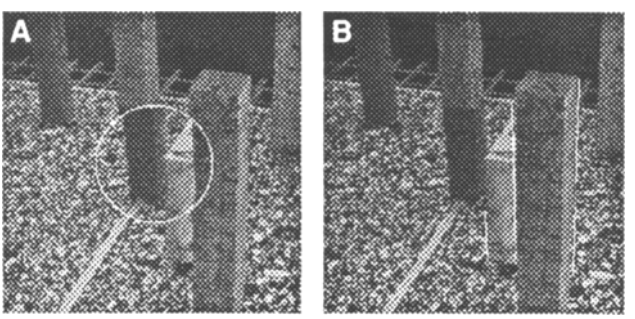

Fig. 5: A: Precise detection of the moving circle obtained from the search for the best circumference approximating the edges of Fig. 4D. Circles with radius between 20 and 80 pixels, with at least half of the points close to an edge and with a local maximum in the number of points close to edges were extracted. The parameters of the circle (center in 130, 129, radius 50 pixels) were correctly estimated. B: Improved detection of the boundaries obtained from the search for the nearly vertical segments of at least 15 pixels better approximating the edges of 4F.