Optical Flow Estimation on Connection-Machine 2

A. Del Bimbo, P. Nesi
Department of Systems and Informatics, Faculty of Engineering
University of Florence, Via S. Marta 3, 50139 Florence, Italy

Abstract

The gradient-based methods for optical flow estimation are based on a constraint equation which is defined for each image pixel. The structure of constraint equation make the problem ill-posed so in the past have been proposed some solutions based on regularization. On the contrary, under the assumption that in the immediate neighbourhood of a pixel the optical flow field is smooth, the constraint equations in that neighbourhood should have a common solution, in this case the problem is not ill-posed. Following this reasoning, an algorithm for evaluating the optical flow, which is suitable for parallel implementation is proposed in this paper. Moreover, parallel implementations of selected algorithms from the literature, for optical flow estimation, are presented in this paper with the intention of comparing their complexity and performance with respect to the proposed approach on a Connection Machine-2.
Index term: computer vision, motion estimation, optical flow, partial differential equation, local voting, parallel implementation, real-time.

1 Introduction

The main problem of sequence analysis in vision is the estimation of the apparent motion usually called "velocity field" or "motion field" [1], which is the perspective projection of the 3D real velocity on the image plane. The estimation of the apparent velocity in a regular grid of the image is useful in solving many problems related to dynamic scene analysis such as 3D motion estimation and 3D object reconstruction, robot navigation, etc..

Recently, the necessity to perform motion analysis in real-time for robot navigation and other applications has provoked an interest in real-time estimation of the apparent motion. Parallel implementation is one way to achieve it. Presently, there are only few examples of parallel implementation for motion estimation in the literature, though, many computational approaches for motion estimation are highly parallelizable.

Three main approaches for solving the motion estimation problem, which are suitable for parallel implementation, can be identified in literature: matching (correspondence), spatio-temporal filtering, and the gradient-based. In the first, local matching techniques are used to evaluate the displacements in subsequent frames for each element of the moving object (lines, corners, patterns) [2], [3]. In the second approach, the estimation of motion is obtained by filtering both in the temporal and frequency domains. Filtering is tuned in frequency and space in order to detect the components of the motion [4], [5], [6], [7]. The third approach (i.e., gradient-based) provides a solution to motion estimation from the observation of brightness changes in the image plane, thus leading to motion estimation of image brightness features [8], [9], [10], [11], [12], [13]. The flow field of these features is normally called "optical flow" of "image flow". The gradient-base approach is suitable for parallel implementation since it requires access only to local image information. An example of is the fully pyramidal implementation presented by Enkelmann in [14]. A parallel implementation on the Connection Machine-2 architecture (CM-2) of the algorithm presented by Tretiak and Pastor in [11] has been presented by Tistarelli in [15], providing quasi-real-time estimations. An example of hardware implementation can be found in Danielsson and al. [16].

The gradient-based approaches evaluate the optical flow by using the so-called Optical Flow Constraint (OFC) equation:

$$E_x u + E_y v + E_t = 0, \qquad (1)$$

where the abbreviation for partial derivatives of the image brightness has been introduced, and u, v correspond to dx/dt, dy/dt, and represent the components of the local velocity vector along the x and y directions, respectively. The definition of the OFC is derived from the observation that the changes in the image brightness $E(x(t), y(t), t)$ are supposed to be stationary with respect to the time variable (i.e., $dE/dt = 0$). In general, boundary and smoothness constraints are needed to obtain a computational solution for the OFC. According to the OFC equation, the optical flow field is defined as the field of image gray value pattern displacements.

Optical flow estimation is susceptible to two main difficulties. The first involves the discontinuities in the local velocity, relating to image brightness discontinuities, which are originated by the presence of noise, too crisp patterns on the moving object surfaces, occlusions between moving objects. Generally, this difficulty can be overcome (or its effect attenuated) by convolving the im-

0-8186-5420-1/93 $03.00 © 1993 IEEE

age with a 2-D or 3-D Gaussian smoothing operator. The second difficulty is the so-called "problem of aperture", which is also present in the human vision [8], and it is related to the impossibility to recover univocally the direction of motion if the object is observed through an aperture which is smaller than the object size.

In this paper, a new algorithm for optical flow estimation in real-time and its parallel implementation is proposed. In addition, new parallel implementations of two well-known algorithms for optical flow estimation, selected from the literature, are presented. All the three algorithms are compared on a SIMD architecture such as the Connection Machine-2. These features of these three algorithms are summarised as follows:

- the approach of Horn and Schunck [8], is a *regularization-based* algorithm, where the optical flow estimation problem, by using the OFC (i.e., $dE/dt = 0$), is considered an ill-posed problem. The solutions are obtained minimizing a functional by means of an iterative process;

- the approach of Tretiak and Pastor [11], is a *multiconstraint-based* algorithm, where the optical flow is obtained by solving a determined system of constraint equations (i.e., $d\nabla E/dt = 0$) at each image pixel;

- the proposed approach is a *multipoint-based* algorithms, which is based on the fact that: if the optical flow changes are smooth, then the OFC equations in a neighborhood of estimation point represent the same velocity, and can be used to define an over-determined system of OFC equations. This approach is derived from the multipoint technique for solving partial differential equation. The obtained over-determined system of equations is solved by using a least-squares technique.

The reference machine used for the parallel implementations of these algorithms is the Connection Machine-2, a SIMD machine with a processing element (PE) for each image pixel, and an efficient communication among processing elements. On such machine all the calculations involved in the algorithms described are performed simultaneously by the PEs assigned to the image pixels.

The paper is organized as follows: Sections 2 presents a parallel implementation of the regularization-based algorithm of Horn and Schunck [8]. A parallel implementation of the multiconstraint-based algorithm proposed by Tretiak and Pastor [11], is discussed in Section 3. In Section 4 an efficient multipoint-based algorithm and its parallel implementation is presented. Comparison of complexity and experimental results for the algorithms discussed is offered in Section 5. Conclusions are drawn in Section 6.

2 Regularization-Based Algorithm

The *regularization-based* approaches consider the optical flow estimation an ill-posed problem [8], [9], [17]. Solutions are obtained minimizing a functional, where a smoothness constraint is appropriately weighted to regularize the solution. Usually, these methods lead to iterative solutions, and the velocity is evaluated in every point of the image. The drawbacks of these approaches are related to the fact that difficulties occur in the presence of object occlusions, and the depth of propagation of the field depends on the number of iterations and on the weighting factor. Horn and Schunck solution [8] is taken as representative of this class of algorithms. The defined functional in this case is:

$$F = \iint \left[(E_x u + E_v v + E_t)^2 + \alpha^2 (u_x^2 + u_y^2 + v_x^2 + v_y^2)\right] dx dy, \qquad (2)$$

where the first term is the OFC, the second term is taken as a measure of the goodness of OFC approximations, and α is a weight factor to control the influence of the smoothness constraint. This functional was minimized by using calculus of variations. That approach leads to a system of two coupled differential equations from the Euler Lagrange equations.
These equations are decoupled and solved iteratively by using the discrete approximation of the Laplacian, whereby a couple of iterative equations is obtained by using a finite difference method. These are then used to estimate the optical flow components:

$$u^{n+1}_{i,j,t} = \bar{u}^n_{i,j,t} - \frac{E_{x i,j,t}\left[E_{x i,j,t}\bar{u}^n_{i,j,t} + E_{y_{i,j,t}}\bar{v}^n_{i,j,t} + E_{t i,j,t}\right]}{\left(\alpha^2 + E^2_{x i,j,t} + E^2_{y i,j,t}\right)},$$
$$v^{n+1}_{i,j,t} = \bar{v}^n_{i,j,t} - \frac{E_{y_{i,j,t}}\left[E_{x i,j,t}\bar{u}^n_{i,j,t} + E_{y_{i,j,t}}\bar{v}^n_{i,j,t} + E_{t i,j,t}\right]}{\left(\alpha^2 + E^2_{x i,j,t} + E^2_{y i,j,t}\right)}, \qquad (3)$$

where: $E_{x i,j,t}$, $E_{y_{i,j,t}}$ and $E_{t i,j,t}$ are estimated by using equations:

$$E_{x i,j,t} = (E_{i+1,j,t} - E_{i-1,j,t})/2,$$
$$E_{y i,j,t} = (E_{i,j+1,t} - E_{i,j-1,t})/2, \qquad (4)$$
$$E_{t i,j,t} = (E_{i,j,t+1} - E_{i,j,t-1})/2,$$

where, only the estimations of $E_{x i,j,t}$, and $E_{y_{i,j,t}}$ require the communication of the pixel data from neighbouring PEs; and:

$$\bar{u}^n_{i,j,t} = (u_{i-1,j-1,t} + u_{i-1,j+1,t} + u_{i+1,j-1,t} + u_{i+1,j+1,t})/12 + (u_{i-1,j,t} + u_{i+1,j,t} + u_{i,j-1,t} + u_{i,j+1,t})/6,$$
$$\bar{v}^n_{i,j,t} = (v_{i-1,j-1,t} + v_{i-1,j+1,t} + v_{i+1,j-1,t} + v_{i+1,j+1,t})/12 + (v_{i-1,j,t} + v_{i+1,j,t} + v_{i,j-1,t} + v_{i,j+1,t})/6,$$

where n is the iteration number. In this iterative solution a guessed value for optical flow estimation at time t can be obtained from the previous time-step (i.e., $\bar{u}^0_{i,j,t} = u^{\omega}_{i,j,t-1}$, where ω is the number of iterations executed at the previous time step).

2.1 Regularization-based: parallel implementation and complexity

The parallel solution of Horn and Schunck algorithm is composed of two phases. In the first estimations of the image brightness derivatives are calculated. The second is an iterative calculation process defined by equations (3). It should be noted that both these phases involve communication of data among neighbouring PEs.

The explicit complexity of this solution is strongly dependent on the number of the iterations, I_t, and results on a sequential machine in $C() = 3M^2 + I_t M^2$ where M is the image dimension; the first term is due to the estimation of the partial derivatives for the image brightness, and the second to the iterative process for evaluating equations (3). On a mesh-connected parallel architecture like the Connection Machine-2 with $M \times M$ processing elements the asymptotical complexity is $O(I_t)$. The number of floating point operations (FLOP) which have to be executed at each iteration by each PE is only 49.

3 Multiconstraint-Based Algorithm

The *multiconstraint-based* approaches for optical flow estimation are based on the observation that the condition $dF/dt = 0$ can be made valid for any motion-invariant function F such as contrast, entropy, average, variance, etc., instead of the image brightness, E, in the OFC. By using a set of these constraints, which are evaluated at the same point in the image, an over-determined set of equations with u and v as unknowns can be obtained [18]. Other methods derive constraint equations which can be regarded as obtained by taking the first derivatives of optical flow constraint with respect to x, y and t [10], [11], [12], [13]. These multiconstraint-based approaches use traditional numerical methods for the inversion or pseudo-inversion of the coefficient matrix of the set of equations. In general, most of the multiconstraint-based algorithms are suitable for parallel implementation on SIMD architectures, since they use only local information.

In this paper, the solution of Tretiak and Pastor [11] has been taken as a representative of the multiconstraint-based direct solutions. In particular, this solution adopts a couple of constraint equations which can be obtained by taking the derivative of the OFC equation with respect to x and y, and neglecting the first-order derivatives (u_x, u_v, v_x, v_y) of the velocity components [16] or through a different path by using the equations of motion components such as in [12]:

$$E_{xx}u + E_{xy}v + E_{xt} = 0, \quad E_{xy}u + E_{yy}v + E_{yt} = 0. \tag{5}$$

The solution is obtained directly through the inversion of the matrix of coefficients H (i.e., the Hessian). Thus, only for those points where the determinant of H ($g_c = E_{xx}E_{yy} - {E_{xy}}^2$) is different from zero a solution of the optical flow is provided.

The components of the optical flow field are estimated is each pixel by using:

$$u_{i,j,t} = \frac{E_{xy_{i,j,t}}E_{yt_{i,j,t}} - E_{yy_{i,j,t}}E_{tx_{i,j,t}}}{g_{c_{i,j,t}}}, \quad v_{i,j,t} = \frac{E_{xy_{i,j,t}}E_{tx_{i,j,t}} - E_{xx_{i,j,t}}E_{yt_{i,j,t}}}{g_{c_{i,j,t}}}, \tag{6}$$

where the second-order partial derivatives of the image brightness are estimate by using the central difference on the first-order derivatives - e.g., $E_{xy_{i,j,t}} = (E_{x_{i,j+1,t}} - E_{x_{i,j-1,t}})/2$. This solution is very sensitive to discontinuities, since it uses the second-order derivatives of the image brightness. To improve the solution quality, an image convolution with a Gaussian filtering was adopted in [11], [16], [15], and also in our experiments.

3.1 Multiconstraint-based: parallel implementation and complexity

The parallel solution to this algorithm is composed of three phases: the first is the convolution of the images with a Gaussian filter, the second is the estimation of the first- and second-order image brightness derivatives, and the third is the estimation of the velocity components from equations (6). The first two phases involve data communication among neighbouring PEs.

The explicit complexity of this solution is strongly dependent on the dimension of the Gaussian pattern which is usually convolved with the image, prior to the optical flow estimation. At any time instant only one new image is filtered, while the optical flow estimation is calculated for the previous sequence. The explicit complexity involved on a sequential machine is $C() = 2F^2M^2 + 3M^2 + 5M^2 + M^2$, where M is the image dimension, and F is the dimension of the Gaussian filtering pattern. The first term of the above equation is due to the Gaussian filtering, the second to the estimation of the first-order derivatives of the image brightness, the

Filtering	F				
	5	7	9	11	13
Time in msec.	22	60	108	183	291

Table 1: Timing expressed in milliseconds for the convolution of the image with a pattern with dimension $F \times F$, on Connection Machine-2 with vp-ratio 1:4.

third to the estimation of the second-order derivatives of the image brightness (E_{xx}, E_{yy}, E_{xy}, E_{tx}, E_{ty}), and the forth to the estimation of the optical flow by using equations (6). The asymptotical complexity on a parallel architecture such as a mesh of $M \times M$ PEs is $O(F^2)$. The number of floating point operations that must be executed by each PE to estimate the optical flow and filter the results are $2F^2+27$. As can be seen in Table 1, increasing the dimension of the filter mask, the filtering stage becomes the dominant (due to the intensive communication required among PEs within the filtering mask boundaries), compared to the calculation effort in image brightness derivatives and optical flow estimation which takes only about 22 msec. on a Connection Machine-2 with 64 floating point unit and a vp-ratio of 1:4.

4 Multipoint-Based Algorithm

Considering that the optical flow changes follow a law which is approximately linear, a smoothed solution for optical flow estimation can be obtained from a linear approximation of the adopted constraint in the neighborhood of the point under consideration. This assumption is valid only if the optical flow field under observation is smooth. Then, the constraints evaluated in a set of neighboring pixels at a certain point represent the same velocity, as a first approximation. This approach was called multipoint, and in literature several cases are presented [19], [20], [13].

In this section, an improved version of the multipoint-based algorithm proposed by Del Bimbo, Nesi, and Sanz in [21] has been presented and parallelized. Considering a generic point $\mathbf{p}$ on the image plane having velocity components (u, v). If the optical flow changes following a law which is approximately linear in x, y, then each point in the neighbourhood of $\mathbf{p}$ has approximatively the same velocity components of $\mathbf{p}$ [21], hence, an over-determined system of $N \times N$ OFC equations:

$$E_{t(i,j,t)} + E_{x(i,j,t)}u + E_{y(i,j,t)}v = 0,$$

can be defined, where N is the dimension of the image neighbourhood around the point $\mathbf{p}$ (if N is odd for $i = -(N-1)/2, .., (N-1)/2, j = -(N-1)/2, .., (N-1)/2$). This over-determined system of $N \times N$ equations has 2 unknowns (i.e., $N \geq 2$), which are the velocity components (u, v) of $\mathbf{p}$. The solution of this over-determined system of equations is obtained by means of a least-squares technique in each estimation point.

Augmenting the neighbourhood dimension, N, around the pixel under consideration smoother optical flow fields can be obtained. On the other hand, large N values lead to a loss in resolution on the moving object boundaries.

The presented algorithm is less sensitive to the discontinuities than the methods which use the second-order partial derivative of the image brightness with the same neighbourhood dimension, N [21]. On the other hand, inaccurate results can be obtained, since the estimation of the optical flow field is computed in a pixel neighbourhood, disregarding the possible difference in velocities. Therefore, it can be used safely only when the optical flow is smooth.

4.1 Multipoint-based: parallel implementation

The multipoint-based algorithms work locally on the immediate neighbourhood of each pixel, and thus can be profitably mapped on a mesh architecture, where a PE is assigned to each pixel of the image. Corresponding pixels of three consecutive images which belong to the same time window used (to estimate the partial derivatives of the image brightness in (4)) are stored in each PE. Thus each PE can directly manages the time history of the corresponding to its pixel (4c).
After the estimation of the image brightness derivatives, each PE (pixel) has an OFC equation. Then, each PE receives from the $N \times N$ neighbouring PEs the coefficients of their OFC equations. Every PE has in this way an over-determined set of $N \times N$ OFC equations in 2 unknowns:

$$A\mathbf{V} + K = 0,$$

where $\mathbf{V}$ is the optical flow vector with components u, v; $A \in \mathcal{R}_{N^2 \times 2}$ matrix of coefficients, with $a_{r,1} = E_{x_r}$ and $a_{r,2} = E_{y_r}$; and $K \in \mathcal{R}_{N^2}$ vector with known terms $k_r = E_{t_r}$ for $r = 1, .., N^2$. An increase in N, the neighbourhood size considered, leads to a significant increase in memory requirements to store the matrix and vector elements at each PE (for instance, with $N = 7$, 147 memory locations are required for storing the coefficients of matrix A and vector K above).

The solution of the over-determined system of equations by using the least-squares technique consists of minimizing the norm:

$$\|A\mathbf{V} + K\|^2.$$

This is performed by using the pseudo-inverse technique transforming the above system of equations into a square matrix of coefficients $\hat{A}$:

$$\hat{A}\mathbf{V} + \hat{K} = 0, \tag{7}$$

where $\hat{A} = A^T A$, and $\hat{K} = A^T K$ (i.e., A^T is the transpose of A). This system of equations can be solved by using traditional techniques such as LU decomposition, Gauss Jordan, etc.. In our case the system (7) is composed of 2 equations in 2 unknowns, and the direct solution was adopted. In particular, the coefficients of the matrix $\hat{A}$ and of the vector $\hat{K}$ are estimated by using:

$$\hat{a}_{i,j} = \sum_{r=1}^{N^2} a_{i,r}^T a_{r,j} = \sum_{r=1}^{N^2} a_{r,i} a_{r,j},$$
$$\hat{k}_i = \sum_{r=1}^{N^2} a_{i,r}^T k_r = \sum_{r=1}^{N^2} a_{r,i} k_r. \quad (8)$$

The estimation of the $\hat{a}_{i,j}$ and $\hat{k}_i$ (for $i = 1,2$; $j = 1,2$) can be performed by accumulating one term at a time, from the r-th neighbouring OFC equation (for $r = 1,..,N^2$), to obtain the final sum, thereby avoiding the need to store the entire set of N^2 OFC coefficients at each PE's memory.
In the process of accumulation an OFC which has an E_t less than a chosen threshold is ignored as an insignificant constraint equation. Also the constraints which have too large values for E_x and E_y are neglected.

4.2 Multipoint-based: complexity

The explicit complexity for the presented multipoint solution on a sequential machine to estimate a velocity vector for each pixel in an $M \times M$ image is $C() = 3M^2 + 3M^2N^2 + 8M^2$, where the first term corresponds to the estimation of the partial derivatives of the image brightness; the second term is due to the least-squares technique for calculating $\hat{a}_{i,j}$ and $\hat{k}_i$ (for $i = 1,2$, and $j = 1,2$); and the third is due to the method for solving the final system of equations (7). As can be seen observing the expresion of $C()$, the asymptotical complexity of the multipoint solution on a sequential machine is of $O(M^2N^2)$.

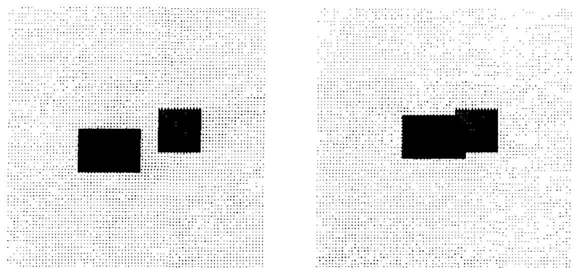
On a parallel architecture, such as the Connection Machine-2, with one PE per image pixel, the asymptotical complexity is reduced to $O(N^2)$, obtaining a respective Speed-Up of about M^2. Table 2 presents the dependence of the execution time of the algorithm on N, the neighbourhood size considered. The number of floating point operations required does not depend on the dimension of the image. For $N = 5$ about 270 FLOP (floating point operations) are made by each PE in estimating the optical flow value. To estimate the optical flow at video rate (25 times per second) the calculations should be completed within $40msec.$, demanding a capability of 6750 FLOPS from each PE for a real-time implementation.

5 Experimental Comparisons

In this section a performance evaluation for the three algorithms discussed in previous sections is provided.

algorithm	N			
	3	5	7	9
Time in msec.	45	122	268	469

Table 2: Execution time expressed in milliseconds and number of operations for the multipoint algorithm depending on the dimension of the neighborhood area dimension N, with vp ratio 1:4.



1 8

Figure 1: Sequence of images where two synthetic objects with superimposed plaid pattern move in opposite directions, (180 and 45 degrees with respect to the X-axis, respectively), (1st and 8th frame, with 128 x 128 image resolution).

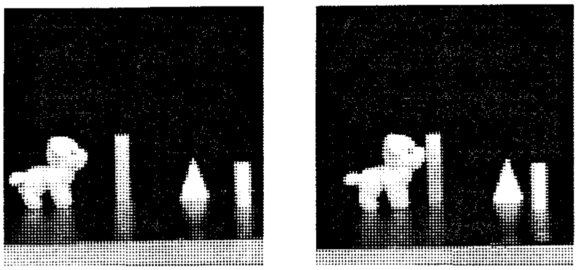
Qualitative comparison of the results obtained by the three algorithms when applied to the test images is offered for selected test cases, along with a comparison of their complexity and the efficiency of their parallel implementation on the Connection Machine-2.

The first test sequence is that of two synthetic objects with a superimposed plaid pattern, which are moving in opposite directions. The plaid pattern consists in the combination of two sinusoidal patterns with orthogonal directions. This sequence was designed to test the performance in the case of occlusion (see Fig.1).

The second test sequence, grabbed from a real environment, presents two moving objects which are moving of translational motion in opposite directions (e.g., the toy dog and the little parallelepipedos). This test sequence has been chosen to test the performance in the case of noisy images (see Fig.2).

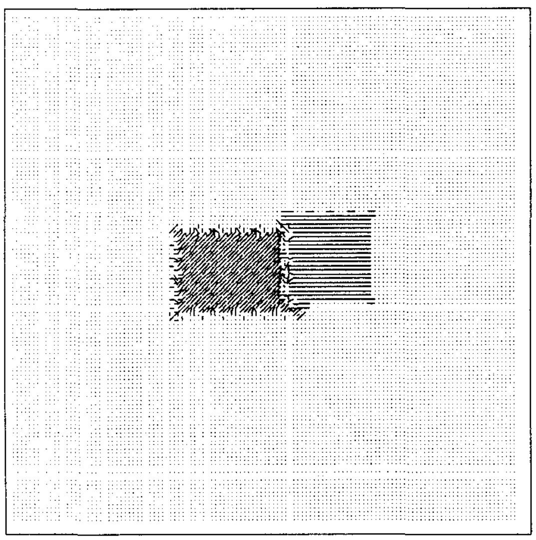
The *multiconstraint-based* solution of Tretiak and Pastor [11], whose parallel implementation has been presented (see Section 3), is highly sensitive to noise, using the second-order derivatives of the image brightness. This is clearly demonstrated in Fig.3. The optical flow estimation obtained with this algorithm is inaccurate at the objects' boundaries, particularly at the boundary between the occluding objects.

The *regularization-based* solution of Horn and Schunck [8], in its parallel implementation presented in Section 4, may produce a smooth optical flow estimation in the presence of noise by increasing the number of iterations, or the α value (see Fig.4). However, doing so has a negative effect at objects boundaries, which are obtained



2 12

Figure 2: Sequence of images where real objects are moving in different directions (2nd, and 12th frame, with 128×128 image resolution).



7

Figure 3: Result obtained by means of the Tretiak and Pastor algorithm (i.e., multiconstraint-based) with respect to the first test case (7th frame), ($F = 9$).

(50) 7

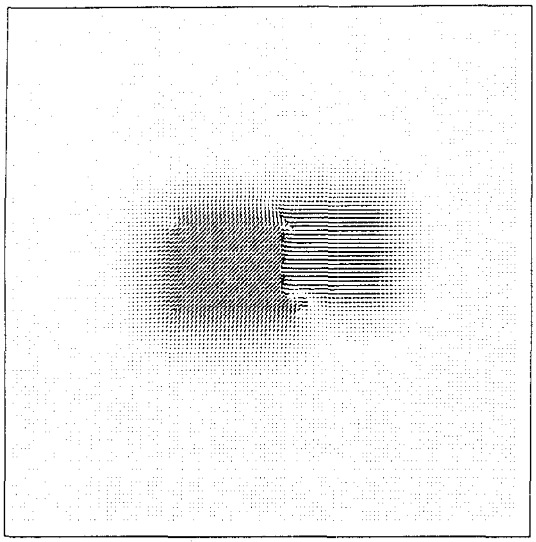
Figure 4: Result obtained by means of the Horn and Schunck algorithm with respect to the second test case (50 iterations, $\alpha = 2.0$), (7th frame).

diffused and with erroneous optical flow estimation (see Fig.5).

For the test case of Fig.2, the *multipoint-based* technique produces an accurate segmentation of the moving objects (sharp boundaries) in image areas where signal to noise ratio is high (see Fig.6).

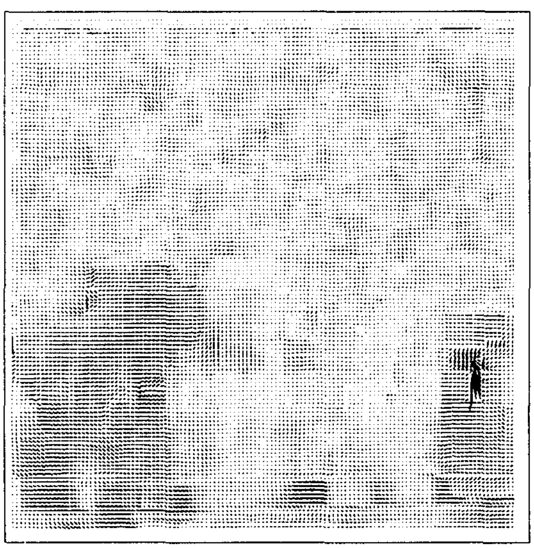
A smoother solution can be obtained with the multipoint algorithm by augmenting the dimension of the neighborhood around each pixel used in the estimation. However, this is also the cause of a loss in resolution at the objects boundaries, and an increase in the computational complexity (see Fig.7). This technique is less sensitive to discontinuities with respect to the multiconstraint-based approach. In the presence of occlusion, at the border between two moving objects, the two objects contribute conflicting velocities, and taking the least-squares estimation yields an inaccurate optical flow estimation, deviating from both.

Table 3 provides a comparison of the algorithms in terms of complexity and efficiency of implementation. For the multipoint-based algorithm real-time performance is obtained, and the complexity is of N^2, with N the dimension of image segment used in the optical flow estimation. The optical flow estimation with Tretiak and Pastor algorithm is obtained in real-time, but overall efficiency is degraded by the need to filter the images with a large Gaussian filter. By using the algorithm of Tretiak and Pastor, it is needed to use large values of F to produce qualitatively comparable results with respect to those obtainable by means of the proposed multipoint algorithm ($N = 5$, $F = 9$). The complexity of Horn and Schunck algorithm is proportional to the number of iterations required to achieve a stable estimation, which is normally of the order of 100 (larger than N^2).



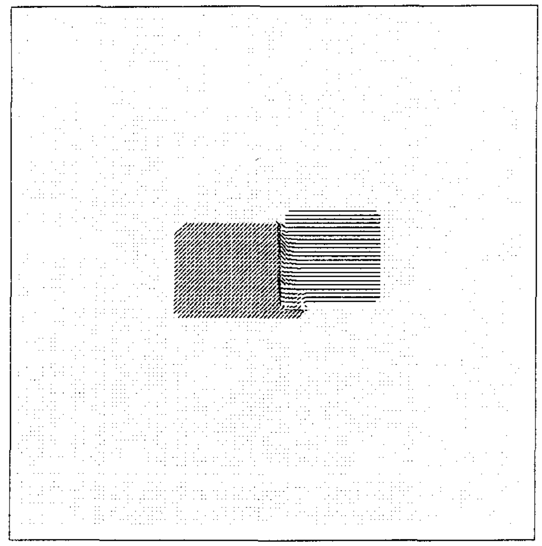
(100) 5

Figure 5: Result obtained by means of the Horn and Schunck algorithm with respect to the first test case (100 iterations, $\alpha = 2.0$), (5th frame).



5

Figure 6: Result obtained by means of the multipoint-based algorithm with respect to the second test sequence with $N = 7$ (5th frame).



7

Figure 7: Result obtained by means of the multipoint-based algorithm with respect to the first test sequence with $N = 5$ (7th frame).

	O()	E_T	Time	FLOP/PE	MFLOPS
M. P.	N^2	120	30	136	19.4
T & P	F^2	130	32.5	108	13.6
H & S	I_t	$15I_t$	$3.75I_t$	$6 + 49I_t$	53.5

Table 3: A comparison of the presented algorithms, in terms of complexity and execution time, E_T – elapsed time (in millisecond) on the Connection Machine-2 with vp-ratio 1:4 (i.e., 4096 PE). The Time column provides the elapsed PE time that could be obtained having one PE per pixel. Parameters used for: the multipoint-based algorithm (M.P.) $N = 5$ and $G = 3$; Tretiak and Pastor algorithm (T & P) $F = 9$.

Referring to the other entries of Table 3, the best performance in terms of number of computations involved (floating point operations - FLOP) is obtained by Tretiak and Pastor algorithm, in its parallel implementation. The Horn and Schunck algorithm requires few FLOPs/PE per iteration, but generally the number of iterations required is large (e.g., $I_t = 100$). The best rate of computations (MFLOPS) is obtained by Horn and Schunck algorithm.

With images produced at a video rate (i.e., 25 frames per second, or 0.04 sec. per frame), real-time performance entails the calculation of optical flow estimation each $40msec$. On the Connection Machine-2 this target is met by the multipoint-based with least-squares estimation algorithm, and by the Tretiak and Pastor algorithm (in the latter, when only the optical flow estimation and a light filtering are considered).

6 Conclusions

A multipoint-based approach for optical flow estimation has been presented together with its parallel implementation. Moreover, parallel implementation of two representative algorithms of the gradient-based approach for optical flow estimation have been presented. All three parallel implementations have been profitably implemented on the SIMD architecture of the Connection Machine-2. Real-time performance has been obtained for the proposed multipoint-based algorithm and for the multiconstraint-based algorithm of Tretiak and Pastor. In terms of qualitative results, it has been found that the multipoint-based is less susceptible to object occlusion and noise than the other algorithms discussed, while the multiconstraint-based algorithm of Tretiak and Pastor has been found the most susceptible to these difficulties.

References

[1] A. Verri and T. Poggio, "Motion field and optical flow: Qualitative properties," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 490–498, May 1989.

[2] J. H. Duncan and T. Chou, "Temporal edges: the detection of motion and the computation of optical flow," in *Proc. of 2nd IEEE International Conference on Computer Vision ICCV'88, Tampa, Florida, USA*, 1988.

[3] H. Bülthoff, J. Little, and T. Poggio, "A parallel algorithm for a real-time computation of optical-flow," *Nature*, vol. 337, pp. 549–553, 9 Febb. 1989.

[4] J. P. H. van Santen and G. Sperling, "Elaborated reichardt detectors," *J. Opt. Soc. Am. A*, vol. 2, pp. 300–321, Feb. 1985.

[5] E. H. Adelson and J. R. Bergen, "Spatiotemporal energy models for the perception of motion," *J. Opt. Soc. Am. A*, vol. 2, pp. 284–299, Feb. 1985.

[6] D. Heeger, "Model for the extraction of image flow," *J. Opt. Soc. Am. A*, vol. 4, pp. 1455–1471, August 1987.

[7] P. J. Burt, C. Yen, and X. Xu, "Multiresolution flow-through motion analysis," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'83, Washington D.C.*, pp. 246–252, 19-23 June 1983.

[8] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[9] H.-H. Nagel, "Displacement vectors derived from second-order intensity variations in image sequences," *Computer Vision, Graphics, and Image Processing*, vol. 21, pp. 85–117, 1983.

[10] R. M. Haralick and J. S. Lee, "The facet approach to optical flow," in *Proc. of Image Understanding Workshop (Science Applications Arlington), Va* (L. S. Baumann, ed.), 1983.

[11] O. Tretiak and L. Pastor, "Velocity estimation from image sequences with second order differential operators," in *Proc. of 7th IEEE International Conference on Pattern Recognition*, pp. 16–19, 1984.

[12] A. Verri, F. Girosi, and V. Torre, "Differential techniques for optical flow," *J. Opt. Soc. Am. A*, vol. 7, pp. 912–922, May 1990.

[13] P. Nesi, A. Del Bimbo, and J. L. Sanz, "Multiconstraints-based optical flow estimation and segmentation," in *International Workshop on Computer Architecture for Machine Perception*, (Paris), pp. 419–426, DGA/ETCA, CNRS/IEF and MEN/DRED, December 1991.

[14] W. Enkelmann, "Investigation of multigrid algorithms for the estimation of optical flow fields sequences," *Computer Vision, Graphics, and Image Processing*, vol. 43, pp. 150–177, 1988.

[15] M. Tistarelli, "Computing optical flow: a real time application of the connection machine system, tr v89-1," tech. rep., Thinking Machines Corporation, DIST Genova, Cambridge, MA, USA, June 1989.

[16] P. Danielsson, P. Emanuelsson, K. Chen, P. Ingelhag, and C. Svensson, "Single-chip high-speed computation of optical flow," in *Proc. of MVA '90 IAPR Workshop on Machine Vision Applications Tokyo*, pp. 331–335, 28-30 Nov. 1990.

[17] B. G. Schunck, "Image flow segmentation and estimation by constraints line and clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 1010–1027, Oct. 1989.

[18] R. J. Woodham, "Multiple light source optical flow," in *Proc. of 3rd IEEE International Conference on Computer Vision ICCV'90, Osaka, Japan*, pp. 42–46, 4-7 Dec. 1990.

[19] C. Cafforio and F. Rocca, "Tracking moving objects in television images," *Signal Processing*, vol. 1, pp. 133–140, 1979.

[20] M. Campani and A. Verri, "Computing optical flow from an overconstrained system of linear algebraic equations," in *Proc. of 3rd IEEE International Conference on Computer Vision ICCV'90, Osaka, Japan*, pp. 22–26, 4-7 Dec. 1990.

[21] A. Del Bimbo, P. Nesi, and J. L. C. Sanz, "Optical flow computation using extended constraints," tech. rep., Dipartimento di Sistemi e Informatica, Facolta' di Ingegneria, Universita' di Firenze, DSI-RT 19/92, Italy, 1992.