



0031–3203(94)00144–8

OPTICAL FLOW BY NONLINEAR RELAXATION

C. COLOMBO, ‡† A. DEL BIMBO* and S. SANTINI*

†Arts Laboratory, Scuola Superiore Sant'Anne, Via G. Carducci 40, 56127 Pisa, Italy

*Dipartimento di Sistemi e Informatica, Università di Firenze, Via S. Marta 3, 50139 Firenze, Italy

(Received 28 July 1993; in revised form 23 September 1993; received for publication 30 November 1994)

Abstract—In this paper, we present a local approach to the computation of raw optical flow fields based on nonlinear relaxation, which is explicitly designed to be coupled with a nonlinear vector filtering technique so as to achieve an effective rejection of the outliers. We derive the approach from a standard linear algorithm—the one-dimensional least squares. This is split down into two steps. First, the linear algorithm is formulated as an iterative procedure, in which image points in a suitable *geometric neighborhood* are made to interact each other. Second, a nonlinearity term in the elementary interaction between neighbors is added, which depends on the Gaussian interaction of elements of a *velocity neighborhood*. The results of a comparison with two linear local techniques show clearly the effectiveness of the approach, both in terms of motion boundary preservation and of optical flow field density.

Optical flow Relaxation methods Optimization Nonlinear filtering

1. INTRODUCTION

The analysis of optical flow fields as a means to infer shape and motion information in an image sequence is one of the most challenging research areas in computer vision, with applications in many different fields, from biomedical engineering to autonomous robots and pattern recognition.

The term *optical flow* refers to a bidimensional velocity field that arises in the image plane, due to local changes in brightness, *under the assumption of an overall constant brightness*.⁽¹⁾ As such, the optical flow is an approximation of the projection onto the image plane of the tridimensional velocity field due to the relative motion of the observer and the observed objects—the so called *motion field*.⁽²⁾ The definition of optical flow above does not have any “operational” meaning, i.e. it does not suggest a procedure for evaluating the optical flow. It determines simply a linear constraint between the two flow field components, that is a *constraint line* in the velocity space, which—apart from the degenerate case of when such a line passes through the origin—is in a 1–1 correspondence with the optical flow component along the direction of the local brightness gradient, or *normal flow*. Computing the optical flow by its definition is thus an underconstrained problem, which exhibits strong analogies with the *aperture problem* of biological vision: The motion of a straight edge observed through a narrow aperture is “undecidable” along the direction of the edge itself.⁽³⁾

Several approaches have been developed in the last few years to solve the aperture problem for the optical

flow by introducing additional constraints and compute, among the infinite possible “optical flows,” a flow both as similar as possible to the actual motion field and suitable for an *a posteriori* interpretation. Typical measures of the performance of optical flow estimation algorithms are the *sensitivity to noise*, the ability in preserving the flow discontinuities (*motion boundaries*) as useful cues for a subsequent segmentation, and the *density* of the flow obtained. The so-called *global approaches*^(1,4) impose an additional constraint on the degree of smoothness of the vector field over the entire image, so that usually they produce an incorrect flattening of the optical flow on motion boundaries. Other techniques have been recently proposed, that explicitly take into account the presence of discontinuities in the motion fields by performing a *piecewise global optimization*.^(5,6) These techniques enjoy the nice feature of simultaneously estimating the flow and segmenting it; yet, they are usually computationally demanding.

Local approaches are among the simplest and fastest for optical flow computing. The flow produced by such approaches is the solution of an overdetermined set of equations, obtained by clustering constraint lines of neighboring points in the image plane^(7,8) or, more generally, by determining more than one constraint at each pixel (see Ref. 9 for a review). By their nature, local approaches do not impose smoothness constraints over large patches of the image, with better results in preserving the flow discontinuities than the global approaches. However, they are likely to produce a noisy optical flow, thereby often requiring a flow filtering step, which is usually not required by global approaches, being already embedded in the flow estimation phase. Figure 1 illustrates the idea behind the

‡ Author to whom all correspondence should be addressed.

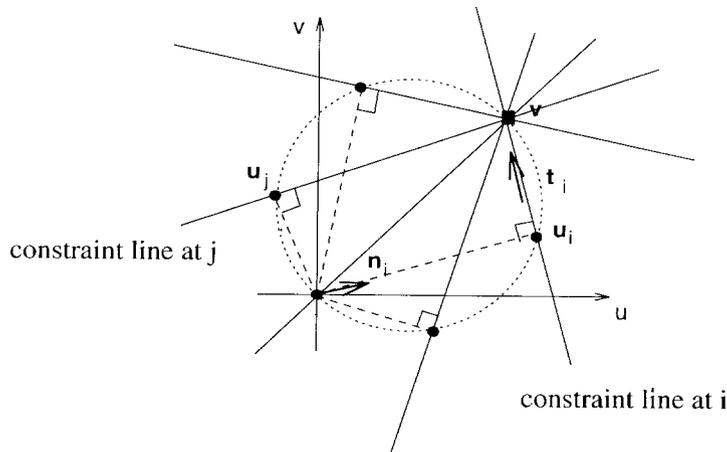


Fig. 1. Optical flow intersection of constraint lines. u (normal flow) is the projection of the optical flow v onto the local brightness gradient direction n (see text).

local approaches, showing the case of a uniform flow field over the entire neighborhood of a generic image point. In this case, the flow can be computed in principle as the unique point of intersection of all the constraint lines in the neighborhood, *provided that the direction of the spatial gradient of brightness is not uniform over the neighborhood.* (This last condition guarantees that in the neighborhood the constraint lines are not all mutually parallel; in which case, no point of intersection could be computed at all, due to the aperture problem.)

In the general case due to non uniform motion inside the neighborhood, to image noise, and the presence of motion boundaries constraint lines have multiple points of intersection,⁽¹⁰⁾ the way to handle this overconstrained problem determines the differences among local approaches. A *pseudo-intersection* of the constraints is obtained as the result of standard least squares procedures, (see for example Refs 7, 11). Such *linear techniques* perform usually well in the presence of noise, but yield unsatisfactory results in the presence of occlusions. Other authors (e.g.^(8,12)) propose *non-linear methods* for constraint clustering, based on “pruning off the outliers” from sets of constraint intersections. These techniques are, in general, especially designed for preserving motion boundaries, and quite sensitive to noise.

In this paper we propose a local approach to optical flow computation, which is based on *nonlinear relaxation*. Relaxation methods have been thoroughly studied in the past, thanks to their ability to solve nonlinear multiconstraint optimization problems⁽¹³⁾ and their amenability to analog VLSI circuits implementation.⁽¹⁴⁾ We derive the approach from a standard linear (one-dimensional least squares) algorithm. This is split down in two steps. First, the linear algorithm is implemented as an iterative procedure, in which image points in a suitable *geometric neighborhood* are made to interact each other. Second, a nonlinearity term in the elementary interaction between neighbors is added, which depends on the Gaussian interaction of

elements of a *velocity neighborhood*. The algorithm is tailored for a combined use with nonlinear smoothing techniques of the raw optical flow. Explicitly, we let the relaxation stage produce a flow field which “decorrelates” at best the noise at its output, so as to be able to filter this long-tailed noise by means of a vector median filter, which is optimal for such kind of noise,⁽¹⁵⁾ and has proven to be effective and fast for performance enhancement of local techniques in general.⁽¹⁶⁾

The logical flow of the paper is the following: in Section 2 we derive the relaxation equations for our method, as the result of the manipulation of the one-dimensional least squares technique. A case study is then presented in Section 3, in which the method is compared with two linear techniques, the one-dimensional least squares (which enables us to emphasize the benefits of the introduction of the nonlinearity term), and the bidimensional least squares technique, a standard technique frequently encountered in applications. The comparison provides us with an insight into the main characteristics of both linear and nonlinear techniques, and clearly points out the superiority of nonlinear, *ad hoc* techniques for the extraction of optical flow fields. We finally draw the conclusions in Section 4.

2. DERIVATION OF THE RELAXATION EQUATIONS

The approach is naturally divided in three different computational phases. In the first phase, the constraint lines and the normal flow are evaluated. A nonlinear relaxation of the normal flow is then performed (second phase), and a raw estimate of the flow field is produced. The final optical flow is then obtained after the third phase of optical flow nonlinear filtering.

This section is organized as follows. In Section 2.1 we derive two alternative forms for the optical flow constraint and define operationally the normal flow. Then, in Section 2.2, we describe in detail the implementation of the one-dimensional least squares technique using relaxation. The nonlinear relaxation term

is introduced in Section 2.3, through heuristical considerations on the interactions among neighboring points at motion boundaries. Finally, in Section 2.4, we describe the nonlinear filtering procedure of vector median filtering of the raw optical flow.

2.1. Formulations of the optical flow constraint

Let us introduce first the canonical form of the optical flow constraint equation.

The optical flow is defined, at each image point (x, y) and at any time t , as the solution $\mathbf{v} = u\mathbf{e}_x + v\mathbf{e}_y$ of the equation

$$\nabla E \cdot \mathbf{v} + \frac{\partial E}{\partial t} = 0, \quad (1)$$

where $\nabla E = (\partial E/\partial x)\mathbf{e}_x + (\partial E/\partial y)\mathbf{e}_y$ is the spatial gradient of the image brightness $E(x, y, t)$.

The constraint line represented by equation (1) determines ∞^1 possible velocity values at a given point p_i , i.e. ∞^1 points (u, v) in the velocity space. Note that, as mentioned in the introduction and shown in Fig. 1, if a number of points with different gradient directions has the same velocity, the relative constraints in the velocity space will be nonparallel lines that meet at a point corresponding to the velocity common to all the points. Note also from the figure that, in the case of shear translation, the locus of all possible normal flow velocities of a neighborhood is a circumference passing through the origin and centered in $(u/2, v/2)$.

The *normal flow* at a given point p_i is expressed as

$$\mathbf{u}_i = (\mathbf{v}_i \cdot \mathbf{n}_i)\mathbf{n}_i = -\left(\frac{\partial E}{\partial t} \frac{\nabla E}{\|\nabla E\|^2}\right)_{p_i}, \quad (2)$$

where $\mathbf{n}_i = (\nabla E/\|\nabla E\|)_{p_i}$ is the unit vector in the gradient direction at p_i . Note that this equation implies that the constraint line can be determined only at points where the image gradient is not zero.

Two alternative formulations of the constraint (1) emerge from the condition that the velocity \mathbf{v}_i at point p_i belongs to the constraint line at p_i . We can express that by rewriting the constraint as

$$\mathcal{E}_i(\mathbf{v}) \doteq (\mathbf{u}_i - \mathbf{v}) \cdot \mathbf{n}_i \quad (3)$$

or, equivalently, by expressing the velocity \mathbf{v}_i as

$$\mathbf{v}_i(s) = \mathbf{u}_i + s\mathbf{t}_i, \quad (4)$$

where $s \in \mathbb{R}$ and the unit vector $\mathbf{t}_i = \mathbf{e}_x \wedge \mathbf{e}_y \wedge \mathbf{n}_i$ provides the local orientation of the constraint line at p_i (refer again to Fig. 1).

2.2. Linear solutions

Two linear solutions of equation (1) at point p_i can be obtained by clustering all the constraint lines in the neighborhood \mathcal{N}_i of p_i , by assuming that the flow be uniform in it, and minimizing the following two quadratic costs (least squares solutions):

$$\mathcal{F}(u, v) = \sum_{j|p_j \in \mathcal{N}_i} [\mathcal{E}_j(\mathbf{v}_i)]^2 \quad (5)$$

and

$$\mathcal{E}(s) = \sum_{j|p_j \in \mathcal{N}_i} [\mathcal{E}_j(\mathbf{v}_i(s))]^2. \quad (6)$$

(These terms are zero if the velocity \mathbf{v}_i belongs to the constraint lines of all the points in \mathcal{N}_i .) In the first case we would have to solve a linear 2×2 system in (u, v) (bidimensional optimization problem) and obtain a solution in closed form. Optimizing equation (6)—which is simply obtained by using the alternative constraint formulation (4) into equation (5), thus imposing that the solution belong to the constraint line—yields the flow vector at p_i as the solution of the one-dimensional optimization problem of finding the right s . By setting

$$\left(\frac{d\mathcal{E}}{ds}\right)_s = 0, \quad (7)$$

we obtain the solution †

$$\tilde{s} = \frac{\sum_j (\mathbf{t}_i \cdot \mathbf{n}_j) [(\mathbf{u}_j - \mathbf{u}_i) \cdot \mathbf{n}_j]}{\sum_j (\mathbf{t}_i \cdot \mathbf{n}_j)^2} \quad (8)$$

and get the flow vector by using the constraint (4)

$$\tilde{\mathbf{v}}_i = \mathbf{u}_i + \tilde{s}\mathbf{t}_i. \quad (9)$$

Notice that the closed form solution can be derived only under the very special circumstances of a quadratic cost.

We show now how to re-formulate the equations in order to solve the same least squares problem by means of an iterative procedure (relaxation). We consider the unknown parameter s as a function $s(\tau)$ of time. ‡ All the velocity values locally interact with each other, looking for a way to mutually satisfy the relative constraints, the relaxation of the constraints being described by a differential equation with \tilde{s} as a stable attractor, and initial condition

$$s(0) = 0, \quad (10)$$

which causes the optical flow be initialized by the normal flow.

The relaxation can be written as a gradient descent method

$$\frac{ds(\tau)}{d\tau} = -\frac{\alpha}{2} \frac{\partial \mathcal{E}(s)}{\partial s}, \quad (11)$$

where the parameter $\alpha \in \mathbb{R}$ controls both the accuracy and the speed of convergence of the successive approximations. Notice that, correspondingly to a displacement ds , the velocity point at p_i is displaced, in the

† The aperture problem would manifest into this equation in the case of insufficient variability of the brightness gradient, by causing the denominator to vanish.

‡ We denote relaxation time by the symbol τ . This emphasizes that relaxation must take place on a different—and much shorter—time scale (adiabatic relaxation) than the movement of the objects in the image. This means that, during relaxation, the objects can be considered as fixed and the values \mathbf{u}_i and \mathbf{u}_j constant.

velocity space, by

$$d\mathbf{v}_i = \frac{d\mathbf{v}_i}{ds} ds = \mathbf{t}_i ds. \tag{12}$$

In the quadratic case of equation (6), equation (11) can be written as

$$\frac{ds(\tau)}{d\tau} = \sum_j \left. \frac{ds(\tau)}{d\tau} \right|_j, \tag{13}$$

where

$$\left. \frac{ds(\tau)}{d\tau} \right|_j \doteq \alpha(\mathbf{t}_i \cdot \mathbf{n}_j) \mathcal{E}_j(\mathbf{v}_i(s)). \tag{14}$$

Equation (14) gives the intensity of the contribution to $d\mathbf{v}_i$ of the constraint line $\mathcal{E}_j(\mathbf{v}) = 0$, with $\mathbf{p}_j \in \mathcal{N}_i$:

$$dv_{ij} = t_i ds|_j. \tag{15}$$

It is easy to show that this contribution will cause a displacement of \mathbf{v}_i in the direction of the intersection between the two constraint lines $\mathcal{E}_i = 0$ and $\mathcal{E}_j = 0$. In fact, the contribution is such that $|\mathcal{E}_j(\mathbf{v}_i(s))|$, the distance between the velocity point \mathbf{v}_i —which belongs to the constraint line of \mathbf{p}_i —and the constraint line of \mathbf{p}_j is minimized:

$$\begin{aligned} \left. \frac{ds(\tau)}{d\tau} \right|_j &= -\frac{\alpha}{2} \frac{d}{ds} [\mathcal{E}_j(\mathbf{v}_i(s))]^2 \\ &= -\alpha |\mathcal{E}_j(\mathbf{v}_i(s))| \frac{d}{ds} |\mathcal{E}_j(\mathbf{v}_i(s))|. \end{aligned} \tag{16}$$

The intensity of displacement (14) will vanish in two different geometrical situations: first, when the two constraint lines are parallel, in which case

$$\mathbf{t}_i \cdot \mathbf{n}_j = 0; \tag{17}$$

second, when the point \mathbf{v}_i corresponds to the intersection of the two constraint lines:

$$\mathcal{E}_j(\mathbf{v}_i) = 0. \tag{18}$$

This last equation confirms that the relaxation process stops (ideally) when all the velocities in the neighborhood of \mathbf{p}_i are at the intersection of all the constraint lines. Besides, equation (17) guarantees that two almost parallel constraint lines—whose intersection in the velocity space is very far—have little interaction.

2.3. Nonlinear relaxation

The linear solutions presented in Section 2.2 leads invariably to blurring motion discontinuities like those one may observe, e.g. in occlusions. To understand this effect, let us consider e.g. the one-dimensional case, and the image projection of two bodies A and B (Fig. 2).

In the velocity space, we have two “stars” of constraint lines. The first star is made of the constraint lines of points of the body A . All these constraint lines converge in a point \mathbf{v}_A , corresponding to the true velocity of A . A second star, corresponding to constraint lines of points of B converges on \mathbf{v}_B .

Let us consider a point \mathbf{p}_i belonging to A and lying very close to the border at which A occludes B . Since we are averaging \mathcal{E}_i , the neighborhood \mathcal{N}_i will contain both points of A and points of B . The velocity point \mathbf{v}_i will move on the constraint line of \mathbf{p}_i under the influence of the other points in A —which will try to pull it towards \mathbf{v}_A —and the points in B —which will try to pull it towards the respective intersections, such as \mathbf{w} in Fig. 2.b. Not only this, but the term $|\mathcal{E}_j(\mathbf{v}_i(s))|$ in equation (16) is bigger for points \mathbf{p}_i belonging to B and therefore—at least initially—the attraction towards \mathbf{w} will dominate the attraction towards \mathbf{v}_A . This will cause \mathbf{v}_i to reach an equilibrium far from \mathbf{v}_A , causing the blurring effect near the borders of the body A . The same reasoning also applies to points on the border of the body B . This is mostly unfortunately, since the borders are the places where the best part of the motion information is contained.⁽¹⁷⁾

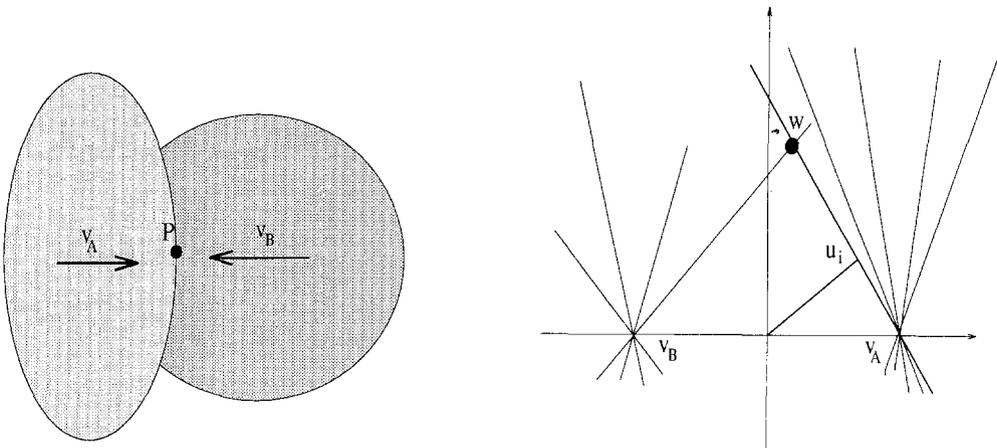


Fig. 2. Occlusion between two bodies with opposite velocities (a), and consequent situation of the constraint lines in the velocity space (b).

The reason for this behavior can be traced to the fact that the least squares solution has no way to make the intersections with constraint lines of A outweigh the intersections with constraint lines of B that are in general much farther away from \mathbf{v}_i . We should introduce a method by which intersections that are close to \mathbf{v}_i are weighted more than the intersections that are far from \mathbf{v}_i .

This is tantamount to the introduction of a *double neighborhood structure* around \mathbf{p}_i . On one side, we have the *geometric neighborhood* \mathcal{N}_i , which dictates that only points geometrically close to \mathbf{p}_i should be considered for relaxation with \mathbf{p}_i . Besides, we introduce a *velocity neighborhood* structure, which dictates that \mathbf{p}_i should weight more, at relaxation time τ , intersections with constraints lines that are closer to the velocity $\mathbf{v}_i(s(\tau))$. The introduction of this second neighborhood structure also introduces a nonlinearity in the relaxation equations. Because of this, a closed-form solution is no longer possible, neither it is possible to trace back to the energy function minimized by the relaxation process.

We suppose that intersection points in the velocity space be modulated by a Gaussian interaction, so that equation (16) rewrites as

$$\left. \frac{ds(\tau)}{d\tau} \right|_j = \alpha(\mathbf{t} \cdot \mathbf{n}_j) \mathcal{E}_j(v_i(s)) \cdot \xi_{ij}(\beta), \quad (19)$$

where

$$\xi_{ij}(\beta) = \exp \left[-\frac{\|\mathbf{v}_i - \mathbf{v}_j\|^2}{2\beta^2} \right]$$

is the Gaussian interaction term that determines the velocity neighborhood structure. Note that since the support of the Gaussian term is the whole real axis, the interaction is never zero. This makes the velocity neighborhood a *fuzzy set*. Notice also that the new nonlinear term is such that $\xi_{ij}(\beta) = \xi_{ji}(\beta)$, that is *the nonlinear interaction between two neighbors in the velocity space is symmetrical*. The parameter $\beta \in \mathbb{R}$ controls the spread of interaction: if β is small, only points that have very close velocity (and that are geometrically close) will interact; if $\beta \rightarrow \infty$, we are back in the linear case, in which all geometrically close points interact, no matter what their velocities are. On the other side, small values of β result in limited relaxation and, therefore, slow convergence.

The presence of the nonlinear term $\xi_{ij}(\beta)$ solves many of the problems presented by the linear solution in handling occlusions. Going back to the example of Fig. 2, we expect that the velocity \mathbf{v}_i be closer, at relaxation time 0, to the velocity of other points of A than to the velocity of points in B ,§ and therefore, we

expect \mathbf{p}_i to interact more with other points in A that with points in B . The closer \mathbf{v}_i gets to the intersection point \mathbf{v}_A , the stronger \mathbf{p}_i will interact with points of A . In the same time, interaction with points of B will become weaker as they start to cluster around the point \mathbf{v}_B . By this mechanism, the term $\xi_{ij}(\beta)$ effectively contributes to the separation of point clusters in the presence of occlusions.

2.4. Optical flow smoothing

A restoration procedure is applied to the raw optical flow field obtained after relaxation, which is specifically aimed to remove the estimation errors made in the relaxation phase due to spikes of noise, and to the presence of motion boundaries passing exactly through the center of the neighborhood. These errors unavoidably occur due to the fact that the raw optical flow vector estimated at a generic image point is assumed to belong to the constraint line at the same point, and this can be even grossly incorrect if the point lies on a motion boundary or if the image noise at that point is big.

For this purpose, vector median filtering—a nonlinear processing—is used in the place of the traditional averaging techniques, thus achieving a better edge response (motion boundaries preservation). In fact, it can be easily shown that the median is an optimal estimate in the maximum likelihood sense if the input distribution is double exponential (longtailed noise), while the average (a linear technique) is the maximum likelihood estimate for the Gaussian distribution (short-tailed noise).

The filtered optical flow vector at \mathbf{p}_i , $\bar{\mathbf{v}}_i$, is computed as the vector median (VM) of the raw optical flow vectors $\bar{\mathbf{v}}_j$ of the neighborhood \mathcal{N}'_i of \mathbf{p}_i

$$\bar{\mathbf{v}}_i = VM(\{\bar{\mathbf{v}}_j, j | \mathbf{p}_j \in \mathcal{N}'_i\}). \quad (20)$$

In the general case, the median of a vector set \mathcal{V} is defined as

$$VM(\mathcal{V}) = \mathbf{v}_i \in \mathcal{V} \mid \sum_{\mathbf{v}_j \in \mathcal{V}} \|\mathbf{v}_i - \mathbf{v}_j\|_L = \min_j \quad (21)$$

where $\|\cdot\|_L$ is a suitably defined norm in the velocity space. The characteristics of vector median filtering are known to depend heavily on the choice of the norm, usually resulting in a trade-off between the performance in edge preservation and in noise removal. The most widely used are the L_2^2 , L_1 and L_2 (Euclidean) norms. It has been found that the L_2 norm is the best for edge preservation, while the L_2^2 norm is the best for medium-tailed noise removal, vector median filtering becoming in this case quite similar to an averaging operation. Using an L_1 norm yields a performance which is intermediate between the two. Vector median filtering (especially with the choice of the Euclidean norm) is quite useful for the task of achieving motion boundary preservation and restoration, since it allows to substitute—by its definition (21)—the incorrect raw optical

§Note that this assumption may fail under very special circumstances, for instance in the presence of very rapid rotation or of very sudden changes in the direction of the gradient. All these circumstances can be seen as limitations of the method, to be overcome either by augmenting the acquisition rate or the density of velocity evaluation points.

flow vectors evaluated at a point on a motion boundary with one of the (possibly) correct raw optical flow vectors in its neighborhood.

Notice that the good edge response characteristics of vector median filters are not achievable by a simple componentwise smoothing of the vector field—that is, by applying a scalar median filter to the field components separately.

3. A CASE STUDY

In this section, we present a case study of estimation based on a sequence of synthetic images, of which the true optical flow (“ground truth”) is known, allowing us to carry out performance evaluations and comparisons.

Each image in the sequence—one sample which is reported in Fig. 3—is 64×64 pixel, with 256 gray levels. The sequence represents the nontrivial case of two partially overlapping squares moving on a translating background. The upper square translates with

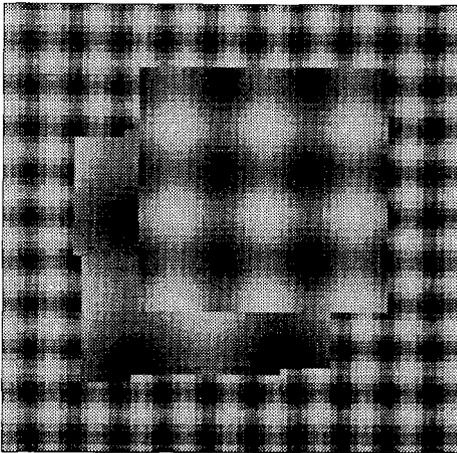


Fig. 3. One frame from the sequence used in the experiment.

a speed of $(-1.2, 1.2)$ pixel/frame. The lower square rotates with a speed of 2.8° /frame. The background translates with a speed of $(-0.707, -0.707)$ pixel/frame (in all drawings, the origin is in the upper-left corner). The squares and the background are covered with a sinusoidal texture, the sinusoid covering the background having a higher spatial frequency.

In this experiment, we estimate the optical flow using three different techniques. We always start with the same normal flow, computed using equation (2) and approximating derivatives by finite differences of image brightness, and use:

- Two-dimensional least squares (2DLS): velocity vectors are not constrained on their constraint lines, and are the solution of the optimization of the cost in equation (5).
- One-dimensional least squares (1DLS): velocity vectors are constrained to lie on their constraint lines, and locally grouped by optimizing the parameter s in equation (4) according to equation (8).
- Our nonlinear relaxation approach (1DNR), which is based on equations (12), (13), (19)—using $\alpha = \beta = 0.1$ —, equations (3) and (10), and is run for 64 iterations.

For all the tests, we report the results both in the image space (x, y) —where velocities are drawn as vectors—and in the velocity space (u, v) —where velocities are drawn as points. The “ground truth” is drawn in Fig. 4, while the normal optical flow is drawn in Fig. 5. The results for the 2DLS algorithm are reported in Fig. 6 (before filtering) and Fig. 7 (after vector median filtering), those for 1DLS are reported in Figs 8 and 9, and those for 1DNR are in Fig. 10 and Fig. 11. Fig. 12 sketches the mean error in all the cases tested.

Let us first comment the results from a qualitatively viewpoint, based on the direct inspection of the figures, and then give and interpret the numerical results. As clear from the figures, most of the error is concentrated near occlusion boundaries, due to the “blurring” effect discussed in the previous section. This is mostly evident

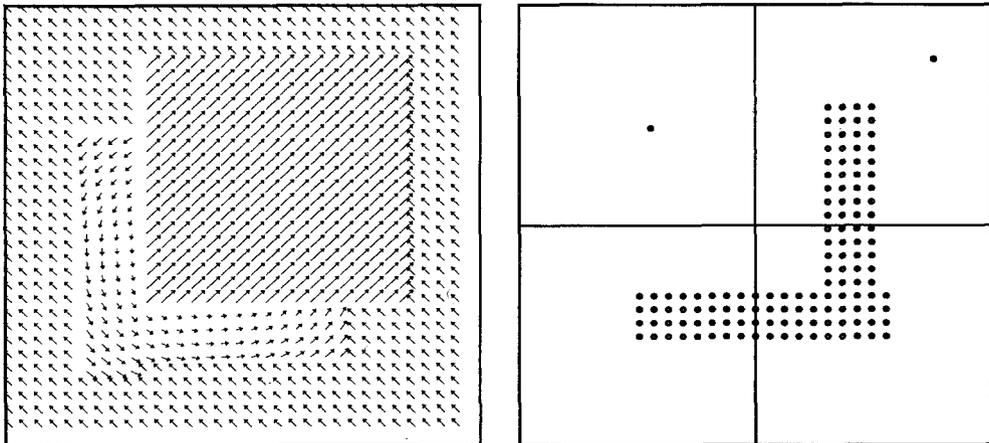


Fig. 4. The “ground truth” for one frame of the experimental sequence.

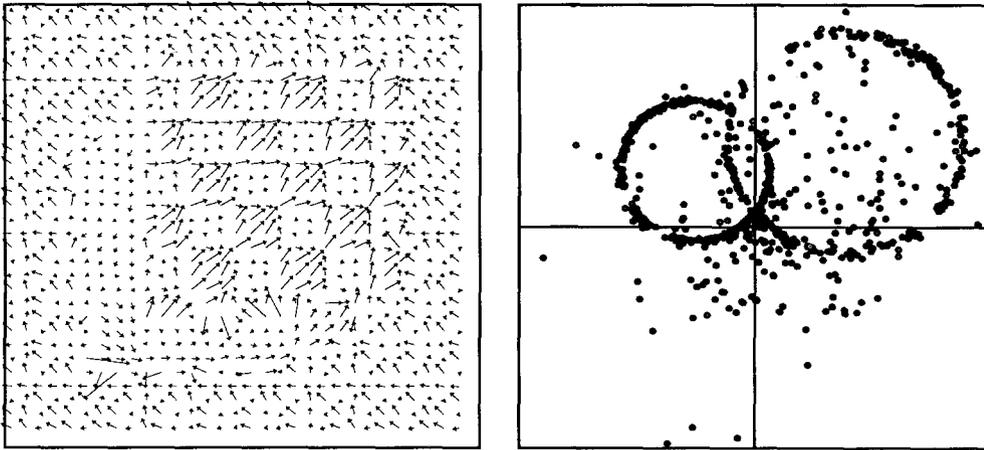


Fig. 5. The computed normal flow. Notice the presence of two circular arrangements of normal flow points, which correspond to the translating objects.

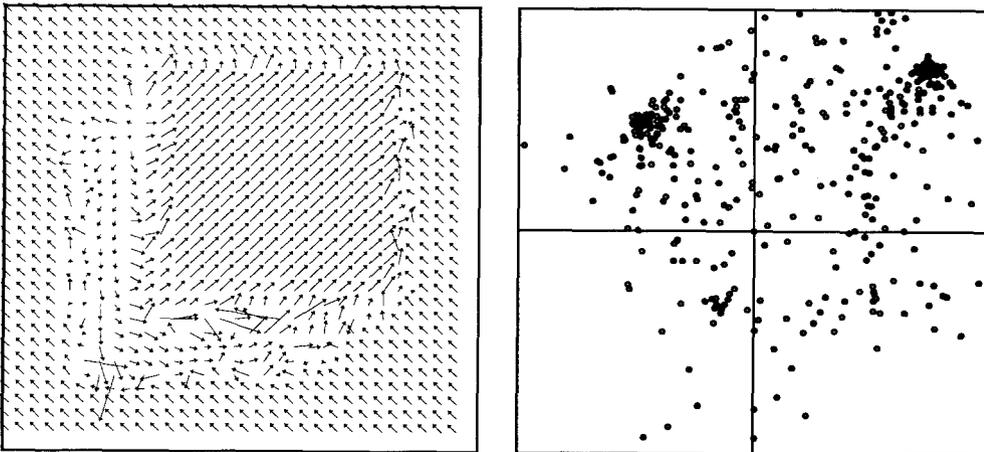


Fig. 6. Optical flow estimation results for 2DLS in (a) the image and (b) the velocity spaces. No filtering.

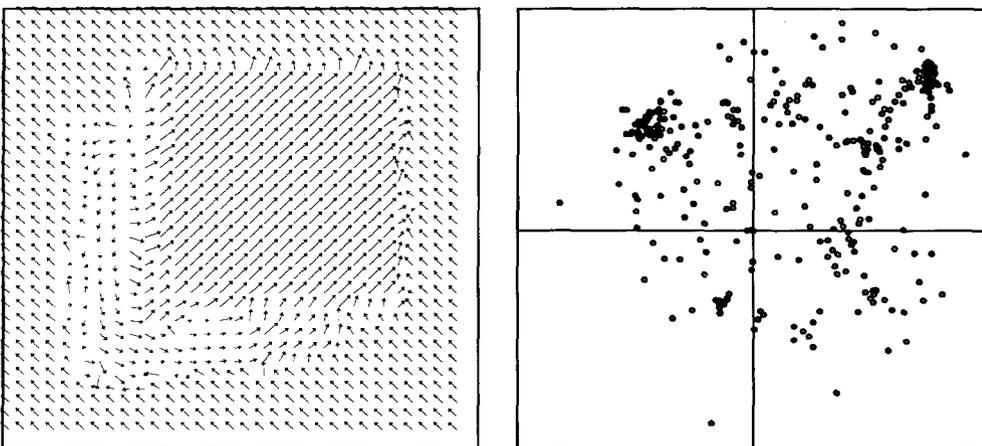


Fig. 7. Optical flow estimation results for 2DLS in (a) the image and (b) the velocity spaces. Vector median filtering.

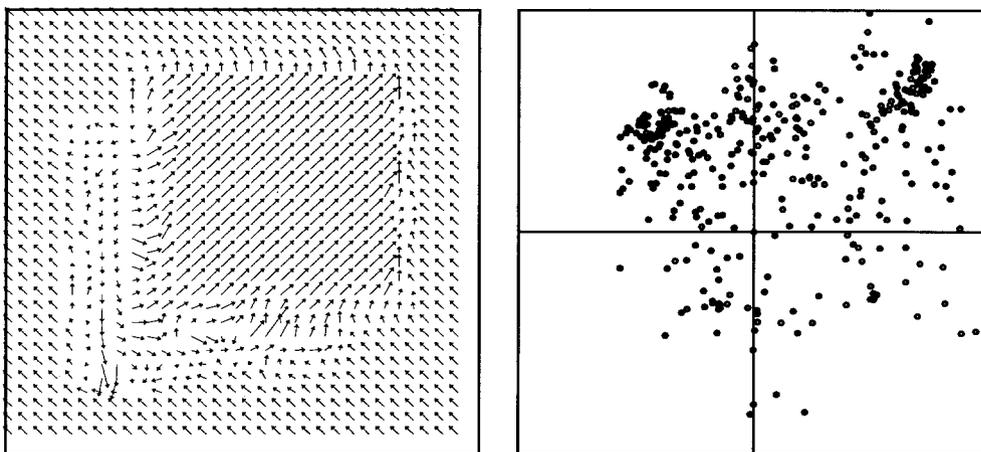


Fig. 8. Optical flow estimation results for 1DLS in (a) the image and (b) the velocity spaces. No filtering.

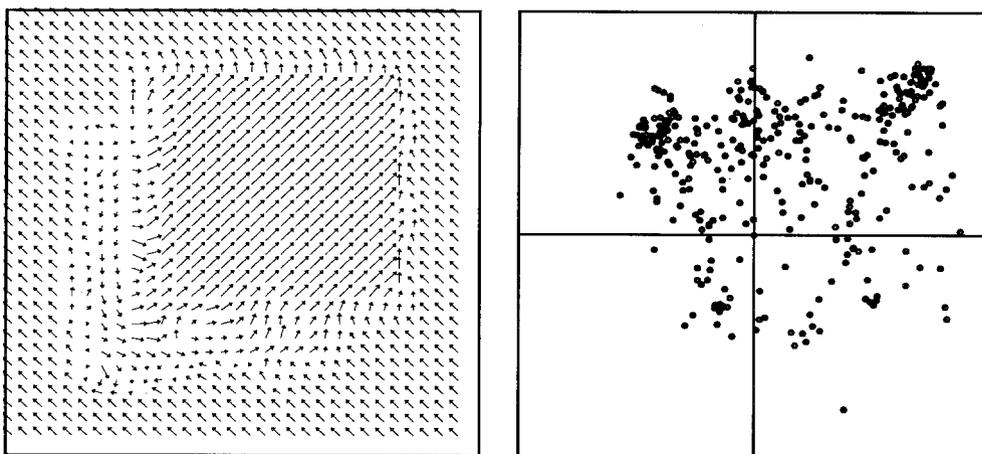


Fig. 9. Optical flow estimation results for 1DLS in (a) the image and (b) the velocity spaces. Vector median filtering.

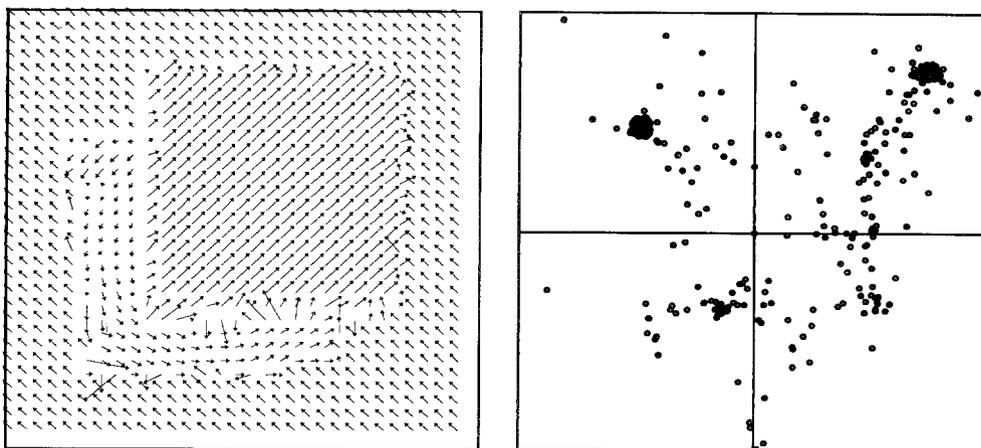


Fig. 10. Optical flow estimation results for 1DNR in (a) the image and (b) the velocity spaces. No filtering.

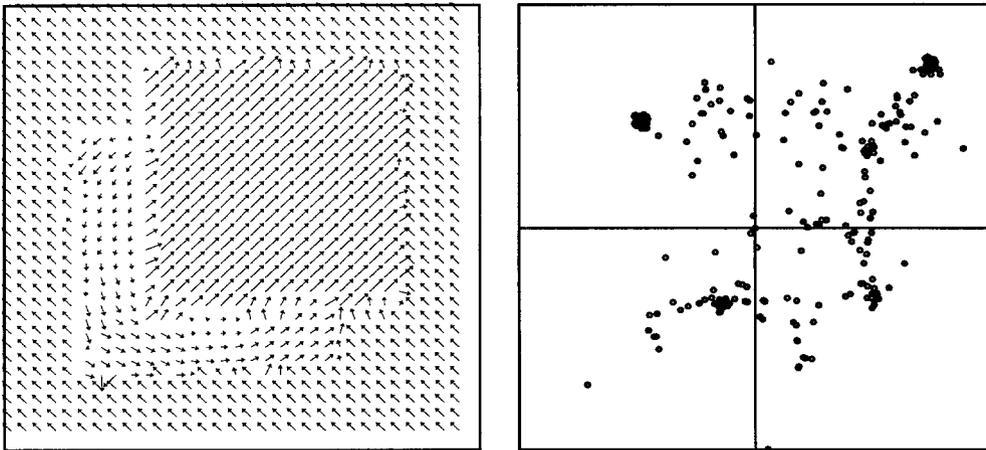


Fig. 11. Optical flow estimation results for 1DNR in (a) the image and (b) the velocity spaces. Vector median filtering.

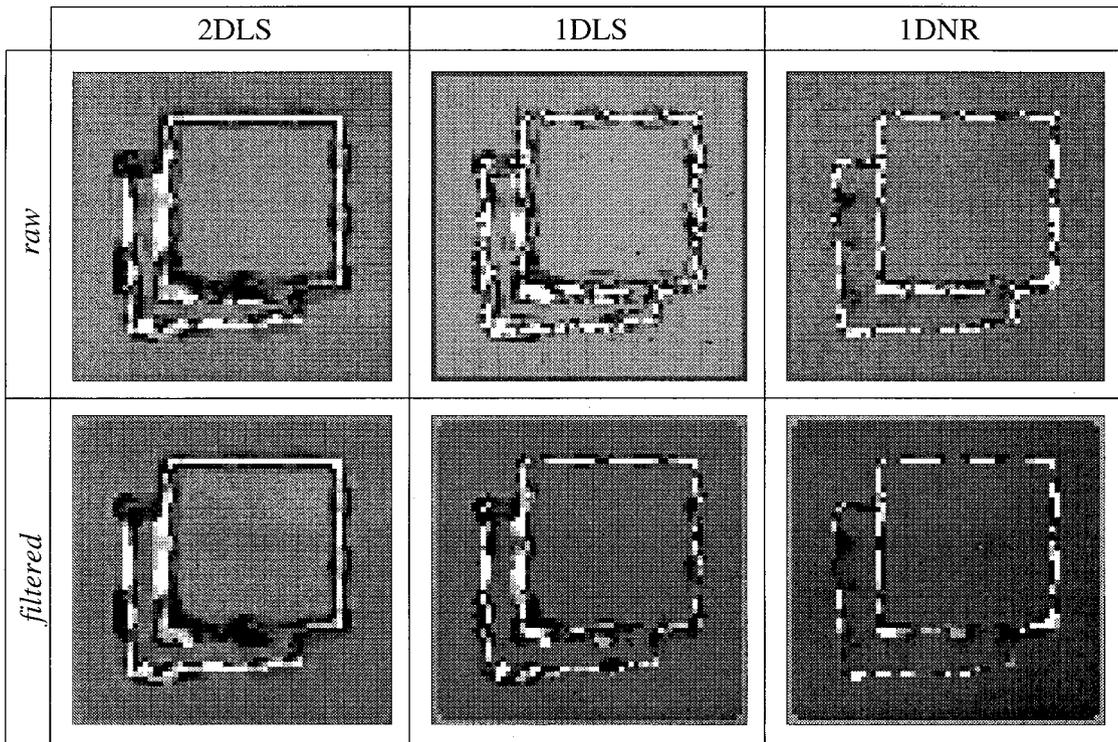


Fig. 12. The distribution of the mean error in the image plane for the three algorithms tested, before and after the filtering step. Brighter areas correspond to bigger errors.

for 2DLS, even after the filtering stage; this is due to the fact that, as mentioned before, vector median filtering is very effective in removing outlier vectors caused by sporadic errors, but not nearly so in removing the highly systematic error caused by 2DLS's blurring. 1DLS represents an improvement in terms of computational effort—it reduces the problem from $2n$ variables to n . Results are worse than 2DLS's before filtering

but, since vector median filtering is more effective in this case than for 2DLS, the final error is lower than for 2DLS. Concerning the 1DNR approach, the figures show that it has, even before filtering, an highly improved behavior in correspondence of the motion boundaries. Yet it exhibits, as we expected, a slightly worst behavior in the interior of the figures. After the filtering step, 1DNR provides the best result of the

three experiments. In fact, it is apparent from Fig. 11b that the two concentrations of points in the velocity space—corresponding to the translation of the upper square and the background—have been correctly identified, and also the “inverted L” of Fig. 4b, characterizing the rotation, has been correctly detected.

We compare now the performance of the three algorithms tested on the basis of the numerical results. Let us denote the “ground truth” by \mathbf{V} , of which the computed optical flow \mathbf{v} is an estimate. The performance parameters are defined as follows: the *local error* is

$$\varepsilon_i = \|\mathbf{V}_i - \mathbf{v}_i\|^2,$$

the *mean error* is

$$\mu = \frac{1}{N} \sum_{i=1}^N \varepsilon_i,$$

and the *rms error* is

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (\varepsilon_i - \mu)^2},$$

where N is the number of pixels in the geometric neighborhood. Finally, we define the *flow density* δ_{μ_0} as the percentage of flow vectors with local error less than μ_0 pixel/frame.

Tables 1 and 2 report the numerical values of the performance parameters μ , σ and $\delta_{0.5}$ obtained in the tests, and Tables 3 and 4 summarize the results of a comparison of the three algorithms. Notice that, *the performance of the algorithms is evaluated both in the overall image and at motion boundaries*, so that it is possible to emphasize the behavior of each technique in the presence of flow discontinuities.

Table 1. Errors with the three algorithms (whole image)

Whole image Parameter	Algorithm					
	2DLS		1DLS		1DNR	
	Raw	Filtered	Raw	Filtered	Raw	Filtered
μ , pixel/frame	0.504	0.483	0.585	0.442	0.511	0.380
σ , pixel/frame	0.541	0.474	1.361	0.440	1.124	0.415
$\delta_{0.5}\%$	76.465	77.222	78.784	81.421	85.352	87.622

Table 2. Errors with the three algorithms (motion boundaries)

Motion boundaries Parameter	Algorithm					
	2DLS		1DLS		1DNR	
	Raw	Filtered	Raw	Filtered	Raw	Filtered
μ , pixel/frame	0.996	0.974	1.253	0.878	1.209	0.774
σ , pixel/frame	0.885	0.829	2.930	0.744	2.975	0.785
$\delta_{0.5}\%$	27.876	27.434	22.124	32.301	33.186	48.673

Table 3. Performance comparison of the three algorithms (whole image)

Whole image Performance	Parameter						
	μ		σ		δ		
	Raw	Filtered	Raw	Filtered	Raw	Filtered	
Best	2DLS	1DNR	1DLS	1DNR	1DNR	1DNR	1DNR
Intermediate	1DNR	1DLS	1DNR	1DLS	1DLS	1DLS	1DLS
Worst	1DLS	2DLS	2DLS	2DLS	2DLS	2DLS	2DLS

Table 4. Performance comparison of the three algorithms (motion boundaries)

Motion boundaries Performance	Parameter						
	μ		σ		δ		
	Raw	Filtered	Raw	Filtered	Raw	Filtered	
Best	2DLS	1DNR	1DNR	1DLS	1DNR	1DNR	1DNR
Intermediate	1DNR	1DLS	1DLS	1DNR	2DLS	1DLS	1DLS
Worst	1DLS	2DLS	2DLS	2DLS	1DLS	2DLS	2DLS

We also analyze separately the results after the raw estimation step and after the vector median filtering step.

We say, of course, that the smaller μ the better the performance, and the bigger $\delta_{\mu\sigma}$ the better the performance. Concerning the performance in terms of rms error though, we consider it to be *inversely proportional to σ before filtering*, and *directly proportional to σ after filtering*. In fact, given the nonlinear filtering scheme adopted—which is mostly effective in the presence of impulsive noise (big values of σ)—a raw estimation algorithm is considered the more effective the more it keeps the noise on the flow field spatially decorrelated.

From the tables it is evidenced that:

- After vector median filtering, the 1DNR has always the best performance.
- 2DLS performs well before filtering, even if only in terms of mean error and not in terms of density. 2DLS becomes the worst of all after filtering, the improvement of performance in the filtering step being very poor. The performance in terms of rms error is also poorer than the two 1D algorithms.
- 1DLS performs always better than 2DLS after filtering, for the reason described above, namely, it allows for a higher spatial decorrelation of the error.
- 1DNR is even better than 1DLS because, for an equal attitude to be nonlinearly filtered, it has a better performance than 1DLS in terms of mean error *before* filtering, being it less sensitive to outliers.

4. CONCLUSIONS

We presented a novel local approach to the estimation of optical flow, which is based on closely coupling nonlinear relaxation with vector median filtering. The nonlinear relaxation technique is obtained as the result of the transformation of a standard linear technique, and is initialized by the normal flow field. We use Euclidean norm for vector median filtering. The case of a “difficult” synthetic sequence was presented, and the knowledge of the “ground truth” enabled us to compare the performance of our method with that of other two local techniques—the one- and bi-dimensional least squares.

The results show clearly the effectiveness of our approach, both in terms of motion boundary preservation and of optical flow field density.

REFERENCES

1. B. K. P. Horn and B. G. Schunck, Determining optical flow, *Artif. Intell.* **17**, 185–203 (1981).
2. A. Verri and T. Poggio, Against quantitative optical flow, *Proc. 1st Int. Conf. Computer Vision ICCV'87*, pp. 171–180 (1987).
3. D. Marr, *Vision*. Freeman, New York (1982).
4. H. H. Nagel, On the estimation of optical flow: relations between different approaches and some new results, *Artif. Intell.* **33**, 299–324 (1987).
5. J. Konrad and E. Dubois, Bayesian estimation of motion vector fields, *IEEE Trans pattern Anal. mach. Intell.* **14**(9), 910–927 (1992).
6. M. J. Black, Recursive non-linear estimation of discontinuous flow fields, *Proc. 3rd European Conf. Computer Vision ECCV'94*, Jan-Olof Eklundh, ed., Springer, Berlin (1994).
7. K. Wahn, L. S. Davis and P. Thrift, Motion estimation based on multiple local constraints and nonlinear smoothing, *Pattern Recognition* **16**(6), 563–570 (1983).
8. B. G. Schunck, Image flow segmentation and estimation by constraint line clustering *IEEE Trans. Pattern Analy. Mach. Intell.* **11**, 1010–1027 (1989).
9. J. K. Aggarwal and N. Nandhakumar, On the computation of motion from sequences of images—a review, *Proc. IEEE* **76**, pp. 917–935 (1988).
10. B. G. Schunck, The image flow constraint equation, *Comput. Vision Graphics Image Process.* **35**, 20–46 (1986).
11. M. Campani and A. Verri, Computing optical flow from an overconstrained system of linear algebraic equations, *Proc. Third IEEE Int. Conf. Computer Vision*, Osaka (1990).
12. D. Ben-Tzvi, A. Del Bimbo and P. Nesi, Optical flow from constraint lines parameterization. *Pattern Recognition* **26** (1993).
13. D. H. Ballard and C. M. Brown, *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ (1982).
14. J. Tanner, C. Mead, Optical motion sensor, *Analog VLSI and Neural Systems*, C. Mood ed., Addison-Wesley, New York (1989).
15. J. Astola, P. Haavisto and Y. Neuvo, Vector median filters, *Proc. IEEE* **78**, 678–689 (1990).
16. F. Bartolini, V. Cappellini, C. Colombo and A. Mecocci, Enhancement of local optic flow techniques. *Time-Varying Image Processing and Moving Object Recognition*, 3, V. Cappellini, ed., Elsevier, New York (1994).
17. O. Braddick, A short-range process in apparent motion, *Vision Res.* **14**, 519–527 (1974).

About the Author—CARLO COLOMBO was born in Bari, Italy, in 1966. In January 1992 he received the laurea degree in Electronic Engineering from the Dipartimento di Sistemi e Informatica of the University of Florence, Italy, where he worked also as a research assistant on dynamic image processing and computer vision. In Fall 1992 he joined the Advanced Robotics Technology and Systems Laboratory (ARTS Lab) of the Scuola Superiore di Studi Universitari e di Perfezionamento S. Anne, Pisa, Italy, where he is currently working towards a Ph.D. in Robotics. His research interests include image processing, pattern recognition, biological and artificial vision, and vision-based control strategies for autonomous robots. He is a student member of the IEEE and the IAPR.

About the Author—ALBERTO DEL BIMBO was born in Florence, Italy, in 1952. He received the doctoral degree in Electronic Engineering from the faculty of Engineering of the Università di Firenze, Italy, in 1977. He was with IBM Italia Spa from 1978 to 1988. Since 1988, he has been Associate Professor of “Digital

Systems” with the Dipartimento di Sistemi e Informatica at the Università di Firenze. He has also been a reviewer for CEC (the Commission of the European Community) at the Directorate of Telecommunications, Information Industries and Innovation, Office and Business Systems Division. From 1994 he has been appointed Full Professor in “Computer Systems.” Professor Del Bimbo is a member of IEEE (Institute of Electrical and Electronics Engineers) and of the IAPR (International Association for Pattern Recognition). He is the Vice-president of the IAPR committee for Industrial Applications, and has been appointed Associate Editor of *Pattern Recognition* journal. His present research interests and activities address imaging technology, including motion analysis, pattern matching, image retrieval from databases and visual languages.

About the Author—SIMONE SANTINI received the laurea degree in Electronic Engineering, and the Ph.D. degree from the Dipartimento di Sistemi e Informatica, University of Florence, Italy, in 1990 and 1994, respectively. Since 1990 he has been visiting the Artificial Intelligence Laboratory at the University of Michigan, Ann Arbor, and the IBM Research Center at Almaden, CA. He is now with the Visual Computing Lab, University of California, San Diego. His primary research interests are cooperative and evolutionary models, and models of the human perceptual system. Dr Santini is a member of the IAPR and the INNS.