Two-dimensional motion analysis

In this chapter, we intend to present several methods of two-dimensional motion analysis, which can be useful to reduce the existing temporal redundancy, as demonstrated in Chapter 1, in an image sequence. This chapter will focus on a short time horizon analysis, often restricted to two images.

First, it is essential to distinguish between apparent motion and real motion (Section 3.1). We shall bring up on the one hand the hypotheses required to obtain a constraint to the apparent motion of a picture point, and on the other hand we shall review the parametric models of velocity fields such as they may result from the geometry of a three-dimensional scene. These two elements of information will be used to estimate motion.

Estimating a velocity or displacement field can be done either in a two-dimensional area, may be possibly the whole image, or in a curvilinear domain, such as a moving contour. Both methods can be used together, insofar as contour analysis is followed by an analysis of the regions. The two approaches will be presented separately, the curvilinear domain in Section 3.2, and two-dimensional domain in Section 3.3. We shall see that in both cases the data observed from an image sequence are similar, and insufficient to locally solve the estimation problem.

In both cases, it is necessary to introduce models of velocity fields which express some spatial consistency, for the estimation problem to be correctly enunciated. Motion field modelization may be deterministic, or stochastic. In the former case, regularization methods are to be used, and in the latter case estimation and filtering methods will have to be used, such as Kalman filter and adaptive filtering.

Also, we address the problems of inter-frame change detection and velocity field segmentation, for which we choose Markovian modelization and Bayesian approach to optimization. For all these fundamental methods (regularization, Kalman filter, adaptive filtering, Markov random field and Bayesian decision), the fundamental notions are reminded in Appendices.

1 2-D velocity field and optical flow

The motion perceived within a sequence of intensity images is not the exact direct projection of the three-dimensional field of velocities. The apparent motion results from a complex process involving, beside real motion, the illumination of the scene, the reflectance properties of the objects composing the scene, the shape of these surfaces, and also the characteristics of the sensor used for image acquisition. It is therefore necessary to make a distinction between real motion, or the two-dimensional field of velocities, and apparent motion, or optical flow. The former results from the geometrical projection of three-dimensional moving objects on the image plane. This implies that geometrical bias has to be corrected by prior calibration. Optical flow can be acquired by observing the spatio-temporal variations of intensity.

First, let us take a geometrical model of two-dimensional velocity field. In Chapter 1, and in case of a perspective projection, we indicated the relation between the two-dimensional coordinates (x, y) of a point according to the point coordinates in three-dimensional space (X, Y, Z)

$$x = \frac{X}{Z} \quad , \quad y = \frac{Y}{Z} \tag{1}$$

If the projection is parallel or orthographic, then the coordinates (x, y) become

$$x = X \quad , \quad y = Y \tag{2}$$

Let $\vec{V} = (\frac{dX}{dt}, \frac{dY}{dt}, \frac{dZ}{dt})$ be the three-dimensional vector of velocity of the point (X, Y, Z). The two-dimensional velocity vector of the point (x, y) is obtained by

$$\left.\begin{array}{l} \frac{dx}{dt} = \frac{1}{Z}\frac{dX}{dt} - \frac{X}{Z^2}\frac{dZ}{dt} \\ \frac{dy}{dt} = \frac{1}{Z}\frac{dY}{dt} - \frac{Y}{Z^2}\frac{dZ}{dt} \end{array}\right\} \tag{3}$$

for a perspective projection. If the projection is parallel, we simply have

$$\frac{dx}{dt} = \frac{dX}{dt} \quad \text{and} \quad \frac{dy}{dt} = \frac{dY}{dt} \tag{4}$$

1.1 Photometric model

On the other side, light intensity variations in time and at any point of the image depend, beside the moving of the objects, on the characteristics of the three-dimensional scene, comprising the surface shape of objects, and the photometric properties of these surfaces. Based on the geometrical optics and photometry, we can state that the light intensity sensed on the image is equal to the radiation emitted from the object surface point towards the sensor, provided that the optical acquisition system has also been correctly calibrated radiometrically [76],

$$I(x, y; t) = L(a, b, \alpha, \beta; t) \tag{5}$$

where (a, b) are the intrinsic coordinates of the surface point, and (α, β) are the polar coordinates of the direction of the point in relation to the surface normal at that point (Figure 1). Equation (5) well illustrates the fact that for any given surface and at any moment, the polar

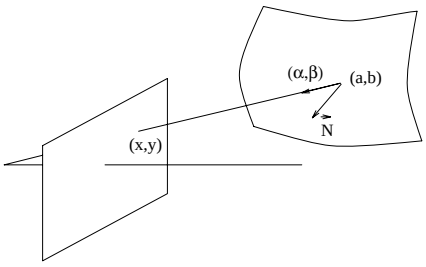

Figure 1: Geometrical model of the scene

coordinates of a point (a, b) will vary with time. This means that in the general situation, the apparent motion does not depend only on the real motion, but also on the three-dimensional geometry of the scene, at least. Let us now consider a Lambertian surface, *i.e.*, a surface on which

$$L(a, b, \alpha, \beta; t) = L(a, b; t) \tag{6}$$

where the light emitted is not direction-dependent. For simplification purposes, we have kept the same light intensity notation in Equation (6). Let us suppose that the scene is evenly illuminated. We can write

$$L(a, b; t) = \rho \vec{I} \vec{N}(a, b; t) \tag{7}$$

where ρ is the albedo of the surface (supposed not to vary spatiotemporally), $\vec{I}$ is a vector which indicates the direction and intensity of illumination, and $\vec{N}(a,b;t)$ is the normal unit vector on the surface, at the point (a,b). Under these hypotheses, Verri and Poggio [76] showed that the velocity vector $(\frac{dx}{dt}, \frac{dy}{dt})$ meets the following equation

$$I_x(x,y;t)\frac{dx}{dt} + I_y(x,y;t)\frac{dy}{dt} + I_t(x,y;t) = \rho\vec{I}\vec{\Omega} \times \vec{N}(a,b;t) \tag{8}$$

where $\vec{\Omega}$ is the angular velocity of the point, (I_x, I_y) is the spatial gradient of intensity, and I_t is the time gradient. Thus the left part of this equation links the velocity 2-D vector to the observed spatio-temporal gradients of intensity in the picture. Conversely, the right part of the equation depends entirely on the 3-D characteristics, which are not easily accessible. It is worth noting that if the hypotheses made are not verified, then additional 3-D characteristics occur in the right part of the Equation (8). Finally, in support of the Equation (8) only in the case of a Lambertian surface evenly illuminated, with a constant albedo at all points, and in a translational motion, we obtain a velocity vector relation independent of the geometrical and photometric characteristics of the 3-D scene.

Restricting ourselves to the observation of intensity spatio-temporal variations, then the equation obtained is known as the optical flow equation (u,v)

$$I_x(x,y;t)u + I_y(x,y;t)v + I_t(x,y;t) = 0 \tag{9}$$

It has to be underscored that in the general situation, optical flow (u,v) is different from the velocity vector $(\frac{dx}{dt}, \frac{dy}{dt})$. It is precisely by comparing the two Equations (8) and (9) that it is possible to spot the difference between the real motion and the apparent motion. An eloquent example of this would be a sphere rotating around a central axis [36]. Under the above hypotheses, the time gradient is null, and therefore a null optical flow vector is compatible with Equation (9). But the real motion is not null and the second part of the Equation (8) is not null either.

In addition, Equation (9) provides a single equation for two unknown values. In fact, this equation permits computing the normal component of optical flow in the direction of the gradient vector $\nabla I = (I_x, I_y)$. Provided that $\| \nabla I \| \neq 0$, we obtain

$$\vec{v}^{\perp} = -\frac{I_t \nabla I}{\| \nabla I \|^2} \tag{10}$$

If the spatial gradient is null and $I_t = 0$, we have no motion-related information on such a point. If the spatial gradient is null, and $I_t \neq 0$, it is a point where the optical flow constraint has not been verified. Such a local impossibility to observe the velocity vector is known as the aperture problem (Figure 2).

The quantity estimated often is a displacement vector $(\Delta x, \Delta y)$, not the velocity vector. An equation is then used, which expresses the invariance of light intensity

$$I(x,y;t) = I(x - \Delta x, y - \Delta y; t - \Delta t) \tag{11}$$

If the Δt interval is small, a first order Taylor development around point $(x,y;t)$ permits obtaining the Equation (9) from (11).

Whichever the Equation (9) or (11), these are insufficient to estimate motion. They express the observations related to the velocity or displacement vectors. Like all observations, they are subject to disturbances. Whether explicitly defined or not, the disturbance model most often used is that of a random additional noise. This noise in a way modelizes the

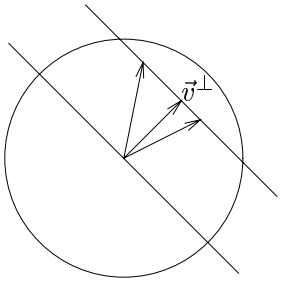

Figure 2: The aperture problem

right part of the Equation (8) which is formed by terms that depend on the 3-D data that are unknown, as long as the 3-D scene reconstruction inverse problem has not been solved. This difficulty justifies modelizing, through a random process, the unknown part of the observation.

1.2 Geometrical model

Let us now return to the geometrical model of the velocity two-dimensional field. Be it $\vec{T} = (T_X, T_Y, T_Z)$ the 3-D vector of translational velocity of the point $P(X, Y, Z)$ and $\vec{\Omega} = (\Omega_X, \Omega_Y, \Omega_Z)$ the angular velocity in relation to the focus at the origin of the axes (0,0,0) (Figure 3). The 3-D velocity vector of point P is given by

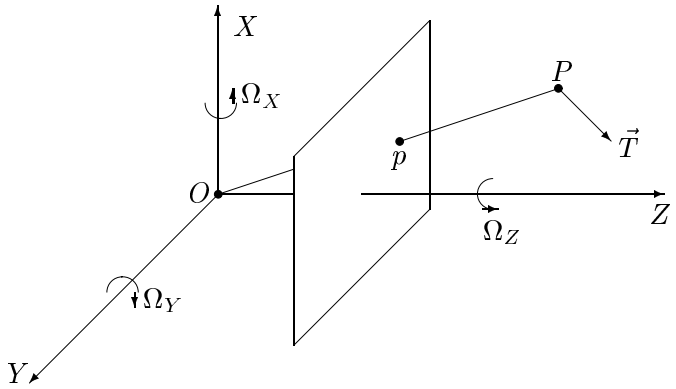

Figure 3: 3-D motion model

$$\begin{aligned} \frac{dX}{dt} &= T_X + Z\Omega_Y - Y\Omega_Z \\ \frac{dY}{dt} &= T_Y + X\Omega_Z - Z\Omega_X \\ \frac{dZ}{dt} &= T_Z + Y\Omega_X - X\Omega_Y \end{aligned} \tag{12}$$

Let us suppose that a surface can be locally approximated by a plane, whose equation is

$$n_X X + n_Y Y + n_Z Z = 1 \tag{13}$$

where the plane hypothetically does not cut the origin. A plane cutting the origin represents a singular case in that it is projected on a straight line of the picture. It can be written from Equation (3.13)

$$\frac{1}{Z} = n_X x + n_Y y + n_Z \tag{14}$$

using the relations (3.1) in case of a perspective projection. The 2-D velocity vector is obtained from Equation (3.3) and with Equations (3.12) and (3.14) as follows

$$\begin{array}{rcl} \frac{dx}{dt} & = & (n_Z T_X + \Omega_Y) + (n_X T_X - n_Z T_Z)x + (n_Y T_X - \Omega_Z)y \\ & & +(\Omega_Y - n_X T_Z)x^2 - (n_Y T_Z + \Omega_X)xy \\ \frac{dy}{dt} & = & (n_Z T_Y - \Omega_X) + (n_X T_Y + \Omega_Z)x + (n_Y T_Y - n_Z T_Z)y \\ & & +(\Omega_Y - n_X T_Z)xy - (n_Y T_Z + \Omega_X)y^2 \end{array} \tag{15}$$

In case of a parallel projection, we can write

$$Z = \frac{1}{n_Z}(1 - n_X x - n_Y y), n_Z \neq 0 \tag{16}$$

The $n_Z \neq 0$ condition means that the plane is not parallel to the Z axis, since in that case it would be projected on a straight line of the picture plane. By using (3.4) and (3.12), we obtain the 2-D velocity vector as follows

$$\begin{array}{rcl} \frac{dx}{dt} & = & T_X + \frac{\Omega_Y}{n_Z} - \frac{n_X}{n_Z}\Omega_Y x - \left(\frac{n_Y}{n_Z}\Omega_Y + \Omega_Z\right) y \\ \frac{dy}{dt} & = & T_Y + \frac{\Omega_X}{n_Z} + \left(\frac{n_X}{n_Z}\Omega_X + \Omega_Z\right) x + \frac{n_Y}{n_Z}\Omega_X y \end{array} \tag{17}$$

It results from the formulas (3.15) and (3.17) that differential spatial equations can be obtained on the 2-D velocity vector. The two equations of the (3.17) formula provide a linear model of the field of velocities in relation to the (x, y) coordinates.

The following relations are then obtained for the velocity vector $\vec{v} = (\frac{dx}{dt}, \frac{dy}{dt})$

$$\frac{\partial^2 \vec{v}}{\partial x^2} = \frac{\partial^2 \vec{v}}{\partial y^2} = \frac{\partial^2 \vec{v}}{\partial x \partial y} = 0 \tag{18}$$

If rotation $\vec{\Omega}$ is neglected, and always for a parallel projection, two first order differential equations are obtained, and a constant model for the velocity field,

$$\frac{\partial \vec{v}}{\partial x} = \frac{\partial \vec{v}}{\partial y} = 0 \tag{19}$$

These spatial differential equations involving the velocity field will be useful for the estimation of that field. In fact, since the local rigidity and planarity hypotheses, which made Equations (3.18) and (3.19) possible to obtain, are not always totally verified, it is equivalent to suppose that the above differential equations are not homogeneous, but controlled by random noises which reflect the model imperfection, whether it is constant or linear.

2 Curvilinear domain

There are a number of reasons for which it would be useful to measure a velocity field in a curvilinear domain, and more specifically in a set of points belonging to a contour which corresponds to surface ruptures of the scene objects. First of all, the velocity field on a contour of a high light intensity variation is very close to the real motion field [76]. This results from the discussion in Section 3.1.1, where the relationship that exists between apparent motion and real motion is presented. Secondly, the spatial detection and localization of the contours provide part of the solution to the problem of 2-D velocity field segmentation. This

is explained in more detail in Section 3.3.3 where this question is addressed. The contour estimated field may latter be used to estimate a dense velocity field in a two-dimensional domain [65]. Finally, it may be of interest in applications where estimation is restricted to a number of points sensitively lower than the overall number of points which compose the picture.

In a curvilinear domain, it is possible to measure only the normal component of the displacement vector of the interested curve. This measurement, used in combination with a displacement field model, permits solving the estimation problem. These two points will be addressed in Sections 3.2.1 and 3.2.2. For a better understanding of what follows, it has to be underlined that, beside the local ambiguity created by the measurement of a single component of the displacement vector, there is also some global ambiguity in certain curves, in as much as the motion field of an entire curve may remain undetermined for certain models of displacement fields. This may be result in a delusional perception and interpretation of the real motion.

2.1 Measurement of the normal component

Detection of moving contours and measurement of the component of the displacement vector perpendicular to the contour may be done jointly [13] [16]. Nevertheless, and for clarity reasons, here these two steps are distinguished.

The initial step, which can be considered as pre-processing, consists in extracting the spatial contours from the intensity picture sequence. This step can be followed by the linking of the detected points, in order to obtain point lists characterized by an overall order, if the contour is open, and additionally by a periodicity, if the contour is closed. The first datum therefore involves the localization and the structuring of the contour points.

The second step is linked to motion. A local operation can only provide a single component of the velocity vector for each point of the contour. This component is perpendicular to the contour. The tangential component to the contour cannot be seen locally. This is a problem of local ambiguity or inobservability. The only exception to this principle is the case of strong curvature points, or points of curve discontinuity, such as corners. It is then possible to measure the two components of the displacement vector locally. The same applies to the extreme points of an open contour. For all these points, measuring the displacement vector requires matching the associated points by means of motion. This pertains to the more general problem of a discrete approach to measure motion by matching visual features [75]. Otherwise, for all other points the normal component of the velocity vector can be measured with Equation (3.10). It is also possible to measure the normal component directly from the detected contours, using geometrical method. We shall remain within that approach to define it further and to analyze measurement errors.

To simplify error analysis, and to remain within general considerations, let us consider a system of axes linked to the point and defined by the normal and tangential directions of the curve at that point. To simplify further, we suppose that the curve can be approximated locally, *i.e.*, in the vicinity of the point, by a circle, whose radius inverse provides the curvature at that point (Figure 4)

$$x^2 + (y - r)^2 = r^2; r > 0 \tag{20}$$

We suppose also that displacement is locally translational. Let (u, v) be the displacement vector. Given the choice of axes, u represents the tangential component and v the normal

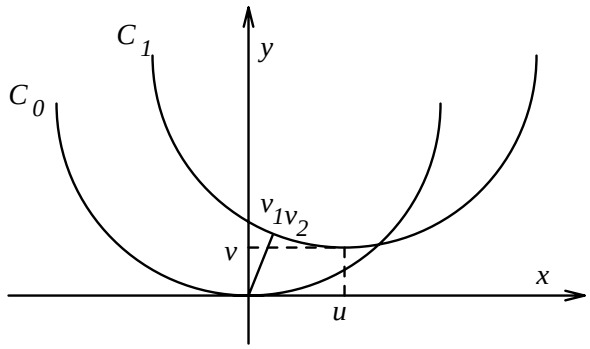

Figure 4: Measurement of the normal component of the displacement vector

component of the displacement vector. The curve at moment t is given by Equation (3.20). We denote C_0 this curve. At moment $t + \Delta t$, this curve has moved to C_1, whose equation is

$$(x - u)^2 + (y - r - v)^2 = r^2 \tag{21}$$

There are two mostly used possibilities to measure the normal component from these two curves: the displacement according to the C_0 curve normal and the distance from the point to the displaced curve C_1 [82]. In Figure 3.4, this corresponds to v_1 (resp. v_2) for the first (resp. second) case. The normal distance v_1 requires one-dimensional search, whereas the distance from the curve requires search in two dimensions. But the second distance (v_2) is always well defined for all the points of the curve, whereas the normal distance is not. This will be better illustrated by the following analysis of measurement errors.

Under the above hypotheses, we can write

$$v_1 = r + v - \sqrt{r^2 - u^2}; r > |u| \tag{22}$$

The normal distance can be defined only if $r > |u|$, which means that curvature must be mild in relation to the inverse of the tangential component of the displacement vector. We also observe that there is a systematic measurement error, which is dependent on the curvature at the considered point and on the tangential component. The error is provided by the following equation

$$\epsilon_1 = v_1 - v = r\left(1 - \sqrt{1 - \left(\frac{u}{r}\right)^2}\right); r > |u| \tag{23}$$

Note that since r is positive, this error is always positive.

Distance v_2 is given by the following equation

$$v_2 = \sqrt{u^2 + (r + v)^2} - r \tag{24}$$

In that case, the measurement error writes as follows

$$\epsilon_2 = v_2 - v = \sqrt{u^2 + (r + v)^2} - (r + v) \tag{25}$$

The distance is now always defined, and the error is always positive, as above. By comparing the two expressions of the error, it is easily demonstrated that, if $v_2 > 0$ (resp. $v_2 < 0$), then $\epsilon_2 < \epsilon_1$ (resp. $\epsilon_2 > \epsilon_1$). Supposing that the probability distribution of the vector (u, v) does not vary with rotation, the result is

$$Pr[v_2 > 0] > Pr[v_2 < 0] \tag{26}$$

The consequence of this is that a greater error is more likely with the normal distance than with the distance to the displaced curve.

Such measurement errors, despite the fact that they are systematic and unavoidable, are in practice ignored or considered as random. Thus a linear relation can be obtained for the measurement equation at each point of the curve. If (n_x, n_y) is the normal vector to the curve, we can write, by definition,

$$n_x u + n_y v = v^{\perp} \tag{27}$$

This equation will be the base for estimating the field of displacement along the curve. Since the displacement is two-dimensional, this equation is insufficient to achieve the estimate. Equation (3.27) is used jointly with a consistency expression of the field of displacement along the curve. We shall present further some estimation methods that we have classified into three categories according to the type of processing : iterative, recursive, or parametric.

2.2 Estimation of the displacement field

Let a list of points belong to the curve C_0, $\{(x(k), y(k)); 0 \leq k < N\}$. If points $(x(0), y(0))$ and $(x(N-1), y(N-1))$ are two adjacent points, the curve is closed. Let $v^{\perp}(k)$ be the normal component of the displacement vector $(u(k), v(k))$ at point $(x(k), y(k))$, and the normal vector of the curve at the same point $(n_x(k), n_y(k))$. Equation (3.27) can write for each point of the curve $(0 \leq k < N)$

$$n_x(k)u(k) + n_y(k)v(k) = v^{\perp}(k) \tag{28}$$

The aim is to estimate the displacement vector $\{(u(k), v(k)); 0 \leq k < N\}$ over the whole set of curve points.

According to the iterative method, all the displacement vectors are estimated globally by successive iterations. By the recursive method, the displacement vector of a point is estimated from that of the preceding point, in which case it is preferable to use a forward recursion and a backward recursion. In the parametric method, the displacement field is assumed to be described by a restricted number of parameters; estimation then amounts to identifying these parameters.

2.2.1 Iterative estimation

Hildreth [34] proposed a regularization method to evade the local ambiguity problem. Some elements of the regularization theory are given in Appendix 3A. This method consists in defining a function of global energy for the field of displacements along the curve. The energy function proposed by Hildreth comprises two terms, and it is quadratic, in relation to the displacement vector components. The first term corresponds to the measurements of the normal component of the displacement vectors over all the points of the curve. The second term expresses a smoothing constraint of the displacement vector field, so that the field be as close as possible to a locally translational displacement.

By using the above notation, the energy function is given by

$$E = \lambda \sum_{k=0}^{N-1} \Big(n_x(k)u(k) + n_y(k)v(k) - v^{\perp}(k) \Big)^2 + \sum_{k=1}^{N-1} \left(u(k) - u(k-1)\right)^2 + \left(v(k) - v(k-1)\right)^2 \tag{29}$$

if the contour is open. If the contour is closed, the following term must be added

$$(u(0) - u(N-1))^2 + (v(0) - v(N-1))^2$$

The weighting factor λ expresses, in statistical terms, the ratio between the smoothing model variance and that of the measurements. Hildreth proposed using the conjugate gradient algorithm to minimize the energy function E which, being quadratic, leads to a system of linear equations. The conjugate gradient algorithm is given in Appendix 3B.

2.2.2 Recursive estimation

In this approach, the correlation of the displacement field along the curve is expressed by an auto-regressive model. Using this model and the measurement of the normal component makes it possible to solve the estimation problem by minimizing the mean squared error. The auto-regressive model used is derived from the model chosen locally for the displacement field. If the displacement field can be considered as a local translation, then we obtain a first order model as follows

$$\begin{bmatrix} u(k) \\ v(k) \end{bmatrix} = \begin{bmatrix} u(k-1) \\ v(k-1) \end{bmatrix} + q(k) \tag{30}$$

where $q(k)$ is a two-component vector which represents the noise of the model. We suppose that this noise is zero-mean, white, and with a covariance matrix $\sigma_q^2 \mathbb{I}_2$, $\mathbb{I}_2$ being the 2×2 identity matrix. This model is equivalent to the smoothing constraint mentioned in the preceding paragraph. Let be a linear model of displacement field in relation to the coordinates. We then obtain the generic expression of the displacement field as follows

$$\begin{aligned} u &= a_{10} + a_{11}x + a_{12}y \\ v &= a_{20} + a_{21}x + a_{22}y \end{aligned} \tag{31}$$

Or we can also write

$$\begin{bmatrix} u \\ v \end{bmatrix} = V_0 + A \begin{bmatrix} x \\ y \end{bmatrix} \tag{32}$$

where the V_0 vector and the matrix A are obviously identified. If the A matrix is null, this model is reduced to that of Equation (3.30). The A matrix takes rotation and some amount of curve deformation into account. A few simple calculations from the (3.31) model provide a third order auto-regressive model [73]

$$\begin{bmatrix} u(k) \\ v(k) \end{bmatrix} = \beta_1(k) \begin{bmatrix} u(k-1) \\ v(k-1) \end{bmatrix} + \beta_2(k) \begin{bmatrix} u(k-2) \\ v(k-2) \end{bmatrix} + \beta_3(k) \begin{bmatrix} u(k-3) \\ v(k-3) \end{bmatrix} + q(k) \tag{33}$$

using the same noise hypotheses as before. By noting $\xi(k)$ as a state vector defined as follows

$$\xi(k) = \begin{bmatrix} u(k) & u(k-1) & u(k-2) & v(k) & v(k-1) & v(k-2) \end{bmatrix}^T \tag{34}$$

we can write

$$\xi(k) = \Phi(k)\xi(k-1) + \omega(k) \tag{35}$$

where $\omega(k)$ is the noise vector induced by $q(k)$ as follows

$$\omega(k) = \begin{bmatrix} q_1(k) & 0 & 0 & q_2(k) & 0 & 0 \end{bmatrix}^T \tag{36}$$

with $q(k) = \left[\begin{array}{cc} q_1(k) & q_2(k) \end{array} \right]^T$. The state transition matrix $\Phi(k)$ is given by

$$\Phi(k) = \left[\begin{array}{cccccc} \beta_1(k) & \beta_2(k) & \beta_3(k) & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \beta_1(k) & \beta_2(k) & \beta_3(k) \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right] \tag{37}$$

The measurement equation of the displacement vector normal component can be written as a function of the state vector as follows

$$v^{\perp}(k) = \left[\begin{array}{cccccc} n_x(k) & 0 & 0 & n_y(k) & 0 & 0 \end{array} \right] + \zeta(k) \tag{38}$$

where $\zeta(k)$ is the measurement noise assumed to be zero-mean and white, with variance σ_ζ^2.

Under all these hypotheses, the estimation problem can be posed as follows : to estimate the displacement field $\{(u(k), v(k)); 0 \leq k < N\}$ over all the curve points, taking the normal displacement values $\{v^{\perp}(k); 0 \leq k < N\}$ into account. The optimization criterion is the minimizing of the mean squared error. This problem is known as that of optimal smoothing, and requires using a Kalman filter in the direction of the contour, and a backward filter to correct the initial estimate by using the information from all the measurements. The Kalman filter and the optimal smoothing are given in Appendix 3C. The Rauch-Tung-Striebel (RTS) formula [63] which expresses optimal smoothing is the only one given here

$$\hat{\xi}(k|N) = \hat{\xi}(k|k) + P(k|k)\Phi^T(k)P^{-1}(k+1|k)(\hat{\xi}(k+1|N) - \hat{\xi}(k+1|k)) \tag{39}$$

where $\hat{\xi}(k|N)$ is the optimal estimate of the state vector using all the observations, $\hat{\xi}(k|k)$ is the optimal estimate using observations 0 to k, as given by the Kalman filter, and $\hat{\xi}(k+1|k)$ is the optimal prediction of $\xi(k+1)$, taking account of observations 0 to k. $P(k|k)$ is the covariance matrix of estimate $\hat{\xi}(k|k)$ and $P(k+1|k)$ is the covariance matrix of prediction $\hat{\xi}(k+1|k)$. These matrices, together with prediction $\hat{\xi}(k+1|k)$, are also produced by Kalman filter. This method was used for model (3.33) by Tziritas [74] and by Rougée *et al.* [65] for model (3.30).

Considering only filtering part and supposing that the measurement noise is predominant on that of the model, and if the model is that of local translation (3.30), then the stochastic gradient algorithm could be used, which is presented in Appendix 3D. We would then obtain

$$\begin{aligned} \left[\begin{array}{c} \hat{u}(k) \\ \hat{v}(k) \end{array} \right] &= \left[\begin{array}{c} \hat{u}(k-1) \\ \hat{v}(k-1) \end{array} \right] \\ &- \epsilon \left(n_x(k)\hat{u}(k-1) + n_y(k)\hat{v}(k-1) - v^{\perp}(k) \right) \left[\begin{array}{c} n_x(k) \\ n_y(k) \end{array} \right] \end{aligned} \tag{40}$$

This method was used by Bouthemy [12]. If the curve is closed, recurrence is cyclical; otherwise, it occurs in forward and backward sense.

2.2.3 Parametric estimation

With a parametric model like that of Equation (3.31), valid for an entire curve or part of a curve, the estimation problem is reduced to estimating the model parameters. This approach was used by Waxman and Wohn [79] with an 8-parameter model, as follows

$$\begin{aligned} u &= a_{10} + a_{11}x + a_{12}y + bx^2 + cxy \\ v &= a_{20} + a_{21}x + a_{22}y + bxy + cy^2 \end{aligned} \tag{41}$$

This model results from the perspective projection of a 3-D space plane on the picture plane, as we have described with Equations (3.15). Therefore, it is only necessary to know the value of the normal component of the displacement vector over more than 8 points of a curve, to obtain a linear equation system that is usually overdefined, and which provides the determination of the 8 parameters by a least-squares method. The conditions for this equation system to accept a single solution are explained in the following section, where the issue of global ambiguity is addressed.

Wohn, Wu and Brockett [82] have used a similar approach involving several parametric models, the most complex being the 8-parameter one (3.41). The difference lies in the use of several iterations to evade the constraint of describing the entire displacement field with only 8 parameters. The displacement field is obtained by successive transformations. The displacement induced by the curve is applied to any obtained solution and the estimation is hence renewed from the curve. Eventually, the displacement is the summation of partial displacements.

2.3 Global ambiguity of the displacement field

There are situations where the displacement field along a curve, as it may result from the measurement of the normal component and of a global displacement model, is not unique. Such a situation is designated as a global ambiguity, well distinct from the local ambiguity involving the normal component at a single point. A clear example is that of a straight line segment in translationali motion. In the absence of additional information relative to the segment ends, there are infinite tangential components that can match the models. The issue of a unique solution may be relevant to applications which aim at reconstructing the motion and structure of 3-D objects, but also when using contour estimates as a reliable base for estimating a dense field over the whole picture.

The ambiguities examined henceforth result from the loss of some motion data due to the projection of 3-D objects on a 2-D plane. Thus the 3-D space curves which are subjected to different motions may appear as moving identically. Another cause can be the difference between real motion and apparent motion, even in 3-D. Thus the resultant projected contour of a rotating sphere may appear immobile.

Human perception of motion may involve several known cases of illusion when interpreting motion [34], such as a rotating ellipse or spiral, or a circular helix on a rotating cylinder. Such illusions result from the choice of one motion interpretation among an infinity of possible interpretations. We shall later present curve shapes that may induce ambiguity in motion interpretation.

Let us now consider the mathematical formulation of the global ambiguity notion. The motion is ambiguous, if at all points of the curve the tangential vector is proportional to a displacement vector

$$\frac{dy}{dx} = \frac{v}{u} \tag{42}$$

If the displacement model is translational, and thus independent of (x, y), it is obvious that the only solution to the differential Equation (3.42) is a straight line.

Let us now consider the 6-parameter linear model (3.31) of displacement field. The following differential equation is obtained

$$\frac{dy}{dx} = \frac{a_{20} + a_{21}x + a_{22}y}{a_{10} + a_{11}x + a_{12}y} \tag{43}$$

Considering the matrix

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \tag{44}$$

and supposing that $\det A \neq 0$. A change of coordinates by translation gives

$$\frac{dy'}{dx'} = \frac{a_{21}x' + a_{22}y'}{a_{11}x' + a_{12}y'} \tag{45}$$

with $x' = x - x_0$ and $y' = y - y_0$, and where

$$A \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} a_{10} \\ a_{20} \end{bmatrix} \tag{46}$$

The solution to the differential Equation (3.45) is known [62]. The results will be given later, using for simplification and to preserve generality, the (x, y) coordinates, and not (x', y'). In any case, any affine transformation a curve that satisfies (3.45) will satisfy a differential equation like (3.43). It is necessary to distinguish the different cases according to the eigenvalues of the A matrix.

2.3.1 Real and distinct eigenvalues

If the eigenvalues of matrix A are real and distinct, $\lambda_1 \neq \lambda_2$, then the solution to the differential equation under consideration gives for the curve the following equation

$$y = cx^{\lambda_2/\lambda_1} \tag{47}$$

Let us take as an example the $y = cx^2$ parabola, or the $xy = c$ hyperbole. In the parabola, vectors like $(u_0, 2cxu_0)$ are tangential at all points of the curve. All types of curves matching Equation (47) are given in Figure 5.

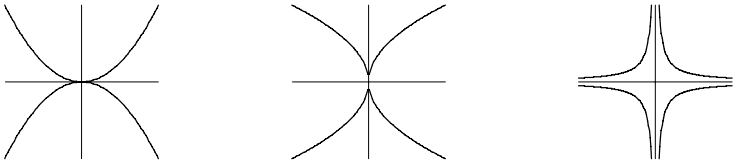

Figure 5: Curves for the cases of real and distinct eigenvalues

2.3.2 Equal eigenvalues

In this case, the curve is given by the equation

$$y = (\gamma + \delta \ln |x|)x \tag{48}$$

This family of curves is illustrated in Figure 6.

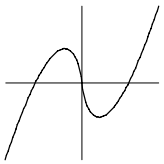

Figure 6: Curves for cases of equal eigenvalues

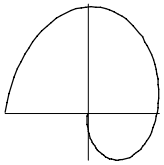

Figure 7: Logarithmic spiral

2.3.3 Complex eigenvalues

The curve equation is given in polar coordinates.

$$\rho = c\exp(a\theta) \tag{49}$$

In the general case, this is a logarithmic spiral equation (Figure 7). If $a = 0$, this is a circle. A linear invertible transformation of a circle produces an ellipse. Tziritas [73] provides examples of 3-D motion parameters leading this type of curve into ambiguous 2-D motions, whether in the general case of logarithmic spiral, or with ellipse or circle.

With regard to the 8-parameter quadratic model (3.41), it is obviously reduced to the linear model, if $b = c = 0$. Therefore, any curve which is ambiguous as to motion interpretation of the linear model, is also ambiguous in the quadratic model. Finally, the increase in model liberty leads other curve families to a situation of multiple interpretation possibilities.

Since these ambiguities are inherent to the motion under consideration, determining a unique solution from the displacement field perpendicular to the curve is not possible, except when reducing the number of degrees of model liberty. For instance, the constant model, which is the simplest and was used in the iterative approach (paragraph 3.2.2), removes the ambiguity from all the curves, except straight line segments. A colinearity test would be sufficient in this model to find out whether a displacement field can be estimated on a given curve. This test amounts to determining the rank of a 2×2 matrix, or to an invertibility test. Likewise, a 6×6 (resp. 8×8) matrix rank test could be used for the linear (resp. quadratic) model produced by Equations (3.33) and (resp. 3.41).

To conclude, except in certain curve families which depend on the motion model used, a displacement field can be estimated on a spatio-temporal contour. This information could be enough in image compression to describe motion, knowing that the displacement field inside a region delimited by occlusion contours is rather continuous and smooth. Thus the motion data are compacted and reduced to the localization of contours and to the description of their displacement fields, as given in the end in a small proportion of picture points [64]. In addition, the displacement field estimated on the contours may constitute a constraint, or initial condition, for the estimation of a dense displacement field at all points of the picture [65].

3 Two-dimensional domain

In this section, we address three major problems of motion analysis in a two-dimensional domain. First, in paragraph 3.3.1, we pose the problem of change area detection based on differences between pictures. Second, in paragraph 3.3.2, we present several methods to estimate the two-dimensional displacement field, which are more or less used for picture sequence compression. In reality, the displacement field is not stationary, which adds to the estimation problem that of the field segmentation, as presented in paragraph 3.3.3.

This chapter will not include the methods of three-dimensional object structure and motion reconstruction, either from the two-dimensional velocity field, or directly from spatio-temporal variations of intensity. This aspect is addressed in Chapter 8.

3.1 Change detection in image sequences

A frequent first step to estimate motion from one picture to another is to detect those parts of the picture whose intensity has changed. This is of interest only when the sensor is stationary, like for instance for videophone or videoconference applications. In fact, it is necessary to distinguish, within the changed part of the picture, that belonging to moving objects and that formed by the background left exposed by the moved objects. Yet this distinction will not be made in what follows, and only the methods used to detect temporal variations of intensity will be exposed. Detection is based on the observation of inter-frame difference

$$D(i,j;k) = I(i,j;k) - I(i,j;k-1) \tag{50}$$

Before presenting the detection methods, it has to be reminded that this quantity was used in the early systems of inter-frame coding [30], known as conditional replenishment. This in fact was a predictive coder, where prediction error was quantized and transmitted provided that it surpasses a certain threshold.

Detection therefore can be achieved by thresholding, which is the simplest manner, or using more sophisticated methods which take the context of the neighborhood of a point into account, to decide on the inter-frame change. A number of studies have been dedicated to that problem [1] [17] [38] [42] [48]. The various techniques are not presented separately in what follows, but different methods are presented in a unified manner. We shall choose a probability modelization of the detection problem, which leads to statistically verify the temporal change hypothesis. Let H_0 be the unchanged pixel hypothesis, and H_1 the change hypothesis. We associate the random field $\Theta(i,j)$ with the events involved in the state of all the pixels of a picture. For each pixel it can be written

$$\begin{array}{ll} H_0: & \Theta(i,j) = 0 \\ H_1: & \Theta(i,j) = 1 \end{array} \tag{51}$$

Let $p(D|0)$ (resp. $p(D|1)$) be the probability density function of the inter-frame difference under the H_0 hypothesis. These probability density functions are supposed homogeneous, *i.e.*, independent of the point (i,j). Most often the variable $D(i,j)$ is supposed to be zero-mean according to the Gaussian or Laplacian laws. The probability density function is provided in the last case by

$$p(D|n) = \frac{\lambda_n}{2} \exp(-\lambda_n |D|); n = 0, 1 \tag{52}$$

Variance is then linked to the λ_n parameter by the relation $\sigma_n^2 = \frac{2}{\lambda_n^2}$.

To determine the hypothesis test, we choose the Bayesian approach. Let us consider three modelization levels for the process $\Theta(i,j)$: point, line or one-dimensional, and region or two-dimensional. The latter two introduce the context notion in the decision process. In this approach, the decision criterion is the maximizing of the *a posteriori* probability (MAP). As to the observations, whichever the modelization level, we suppose that the observation noise is zero-mean and white.

3.1.1 Decision at the point level

The decision strategy that we use requires that the probabilities of two hypotheses are known in advance. Let P_0 (resp. P_1) be the *a priori* probability of hypothesis H_0 (resp. H_1). The absence of *a priori* knowledge is expressed as $P_0 = P_1$.

Under all the assumptions, and using the maximum *a posteriori* criterion, the decision test is as follows:

$$\text{Decide} H_1, \quad \text{if} \quad Pr[H_1|D(i,j)] > Pr[H_0|D(i,j)] \tag{53}$$

With Bayes theorem, it can be written

$$Pr[H_n|D(i,j)] \propto Pr[H_n]p(D(i,j)|H_n); n = 0,1 \tag{54}$$

Let us consider the case where $D(i,j)$ is a Laplacian random variable. Then (3.53) and (3.54) provide the following relation

$$\text{Decide} H_1, \quad \text{if} \quad \frac{P_1\lambda_1}{2}\exp(-\lambda_1|D(i,j)|) > \frac{P_0\lambda_0}{2}\exp(-\lambda_0|D(i,j)|) \tag{55}$$

from which a threshold test is obtained

$$\text{Decide} H_1, \quad \text{if} \quad |D(i,j)| > \frac{\ln P_0\lambda_0 - \ln P_1\lambda_1}{\lambda_0 - \lambda_1} \tag{56}$$

If $P_0 = P_1$, this test is equivalent to the maximum likelihood test, as can be observed with the relation (3.54). In the end, the maximum likelihood test is

$$\text{Decide} H_1, \quad \text{if} \quad |D(i,j)| > \frac{\ln \lambda_0 - \ln \lambda_1}{\lambda_0 - \lambda_1} \tag{57}$$

If $D(i,j)$ is a Gaussian variable, it can be demonstrated in the same way that the maximum *a posteriori* probability test is

$$\text{Decide} H_1, \quad \text{if} \quad |D(i,j)| > \sigma_0\sigma_1\sqrt{\frac{2\left(\ln\frac{P_0}{\sigma_0} - \ln\frac{P_1}{\sigma_1}\right)}{\sigma_1^2 - \sigma_0^2}} \tag{58}$$

where σ_0^2 (resp. σ_1^2) is the variance of variable $D(i,j)$ under the H_0 (resp. H_1) hypothesis.

Like in all two-hypothesis statistical tests, there are two types of error: false detection and non-detection. The above tests presume that these two types of error carry the same cost. The two types of error can be considered distinctly in an approach other than the Bayesian one, which is that of the likelihood ratio. This approach does not require knowing *a priori* probabilities (P_0 and P_1), but only the parameters of one of the two probability densities. The probability of false detection is set at the chosen level, be it α. The likelihood ratio provides sufficient statistics for the test. Let us consider the case of a Laplacian variable;

then the sufficient statistics is the average absolute value of the inter-image difference. The test is still a threshold one. Let s be the threshold; it is set by the following relation

$$Pr[|D(i,j)| > s|H_0] = \alpha \tag{59}$$

With a Laplacian variable, the following equation ensues

$$2\int_s^\infty \frac{\lambda_0}{2} e^{-\lambda_0 x} dx = \alpha \tag{60}$$

This eventually leads to set the threshold according to the false detection probability and to λ_0,

$$s = \frac{1}{\lambda_0} \ln \frac{1}{\alpha} \tag{61}$$

We see that the threshold depends solely on the variance of the no-moving background, which is linked to the sensor noise, not to the moving object characteristics.

Also, the test becomes more effective, if the decision is based on observing the inter-image difference in some amount of the point surroundings, rather than on the point alone. Back within the method of maximum *a posteriori* probability. The decision test is as follows

$$\begin{aligned} \text{Decide} H_1, \text{if} & Pr[H_1|D(i+k,j+l) : (k,l) \in S] \\ & > Pr[H_0|D(i+k,j+l) : (k,l) \in S] \end{aligned} \tag{62}$$

where the S set defines the neighborhood of a point and includes (0,0). An example of such a set is: $S = \{(k,l) : k^2 + l^2 \leq 2\}$ which includes 9 points. Let n be the cardinal of the S set. Following the same approach as that followed for a single point, we obtain the following test, in the case of a Laplacian variable

$$\begin{aligned} \text{Decide} H_1, \text{ if } & P_1 \left(\frac{\lambda_1}{2}\right)^n e^{-\lambda_1 \sum_{(k,l)\in S} |D(i+k,j+l)|} \\ & > P_0 \left(\frac{\lambda_0}{2}\right)^n e^{-\lambda_0 \sum_{(k,l)\in S} |D(i+k,j+l)|} \end{aligned} \tag{63}$$

Finally the following test is obtained

$$\text{Decide} H_1, \text{ if } \frac{1}{n} \sum_{(k,l)\in S} |D(i+k,j+l)| > \frac{\ln \lambda_0 - \ln \lambda_1 + \frac{1}{n} \ln \frac{P_0}{P_1}}{\lambda_0 - \lambda_1} \tag{64}$$

We observe that the role of *a priori* probabilities is the lesser as the number of points considered is larger. In exactly the same manner, the following tests is obtained in the case of a Gaussian variable

$$\text{Decide} H_1, \text{ if } \frac{1}{n} \sum_{(k,l)\in S} (D(i+k,j+l))^2 > \sigma_1^2 \sigma_0^2 \frac{\ln \sigma_1^2 - \ln \sigma_0^2 + \frac{2}{n} \ln \frac{P_0}{P_1}}{\sigma_1^2 - \sigma_0^2} \tag{65}$$

In the Gaussian case, we also give the likelihood ratio test, which only requires knowing the variance σ_0^2, which depends on the camera noise. This is a threshold test, the threshold being set determined by setting the false detection probability. The sufficient statistics is equal to the sum of the squares of the inter-image differences over all the points defined by S. This leads to the following test [1]

$$\text{Decide} H_1, \text{ if } \frac{1}{\sigma_0^2} \sum_{(k,l)\in S} (D(i+k,j+l))^2 > s \tag{66}$$

The threshold s is determined by setting the false detection probability α using the following relation

$$Pr[\frac{1}{\sigma_0^2}\sum_{(k,l)\in S}(D(i+k,j+l))^2 > s|H_0] = \alpha \tag{67}$$

Under the H_0 hypothesis, the variable on which the test is based follows the χ^2 distribution with n degrees of freedom, whose distribution function is tabulated and is used in the threshold determination.

Another way to take the neighborhood into account consists to consider the binary process classifying a point as belonging or not to a changed area. We have already evoked the possibility of a one- or two-dimensional context, which will be presented successively hereafter.

3.1.2 Decision at the line level

We suppose that the $\Theta(i,j)$ process is not independent from one point to the other, but that there is a surrounding relation modelized by a first order Markovian chain. We suppose therefore that

$$Pr[\Theta(i,j) = \theta(i,j)|\theta(i,l); l < j] = Pr[\Theta(i,j) = \theta(i,j)|\theta(i,j-1)] \tag{68}$$

Then the process that describes the fact that the points belong to the changed or unchanged areas is characterized by four probabilities: P_{00}, the probability that a point is unchanged, knowing that the preceding point is also unchanged, P_{01}, the probability of a transition from a changed to an unchanged point, and with a respective interpretation of probabilities P_{10} and P_{11}. The maximum *a posteriori* probability method (MAP) consists in maximizing, at all the points of a given line, the following probability

$$Pr[\Theta(i,j) = \theta(i,j)|D(i,j) = d(i,j); j = 0,1,\cdots,N-1] \tag{69}$$

This conditional probability takes into account all the inter-image difference observations made all along the length N of the line i. Applying Bayes theorem, we obtain the maximized quantity of formula (3.69) as equivalent to the minimized following quantity

$$\begin{aligned}
&-\ln p(d(i,j)|\theta(i,j); 0 \le j < N) - \ln Pr[\theta(i,j); 0 \le j < N] \\
&\quad = -\sum_{j=0}^{N-1} \ln p(d(i,j)|\theta(i,j)) - \sum_{j=0}^{N-1} \ln Pr[\theta(i,j)|\theta(i,j-1)]
\end{aligned} \tag{70}$$

Under the Laplace law hypothesis (52) we obtain the following quantity to be minimized

$$\begin{aligned}
&\sum_{j=0}^{N-1} (\theta(i,j)(\lambda_1|d(i,j)| - \ln\lambda_1) + (1-\theta(i,j))(\lambda_0|d(i,j)| - \ln\lambda_0)) \\
&\quad - \sum_{j=0}^{N-1} ((1-\theta(i,j))\theta(i,j-1)\ln P_{01} + (1-\theta(i,j))(1-\theta(i,j-1))\ln P_{00} \\
&\quad + \theta(i,j)\theta(i,j-1)\ln P_{11} + \theta(i,j)(1-\theta(i,j-1))\ln P_{10})
\end{aligned} \tag{71}$$

Minimization is done in relation to the N unknown values $\{\theta(i,j); j = 0, 1, \cdots, N-1\}$, which may be equal to 0 or 1. The above quantity can also be written as

$$\sum_{j=0}^{N-1} (\theta(i,j)(\lambda_1 |d(i,j)| - \ln \lambda_1 P_{11}) + (1 - \theta(i,j))(\lambda_0 |d(i,j)| - \ln \lambda_0 P_{00})$$

$$\theta(i,j)(1 - \theta(i,j-1)) \ln \frac{P_{11}}{P_{10}} + (1 - \theta(i,j))\theta(i,j-1) \ln \frac{P_{00}}{P_{01}}) \tag{72}$$

This leads to the minimization of

$$\sum_{j=0}^{N-1} \theta(i,j) \left(\frac{\ln P_{00}\lambda_0 - \ln P_{11}\lambda_1}{\lambda_0 - \lambda_1} - |d(i,j)| \right) + \frac{c_{10} \ln \frac{P_{11}}{P_{10}} + c_{01} \ln \frac{P_{00}}{P_{01}}}{\lambda_0 - \lambda_1} \tag{73}$$

where c_{10} (resp. c_{01}) is the number of transitions from the unchanged (resp. changed) to the changed (resp. unchanged) state [17]. It is worth noting that in the absence of any surroundings constraint, *i.e.*, if $P_{00} = P_{01} = P_0$ and $P_{10} = P_{11} = P_1$, we obtain exactly the same result as when deciding independently at each pixel (3.56). It has to be noted also that the larger the ratios P_{11}/P_{10} and P_{00}/P_{01} are, the fewer transitions are detected. Knowing that the addresses of these transitions have to be coded in some way, we appreciate the interest of the Markovian modelization and of the Bayesian approach. So far, we have provided the criterion to be optimized for the detection of inter-frame changes. With regard to resolution methods, there are two possibilities. Minimizing the sum in (3.72) is achieved by using Viterbi algorithm [17] [26]. Otherwise, by using a method proposed by Askar and Derin [3], which gives directly the *a posteriori* probabilities of formula (3.69).

3.1.3 Decision at the two-dimensional level

Finally, let us consider the possibility of a two-dimensional neighborhood, hence the modelization of the $\Theta(i,j)$ field using a Markov random field. It has been shown that a Markov random field is equivalent to a Gibbs distribution. The definition and fundamentals of Markov random fields are given in Appendix 3E. Let us consider an 8-point neighborhood. We can write, for a frame of M lines and N columns

$$\ln Pr[\Theta(i,j) = \theta(i,j); 0 \leq i < M, 0 \leq j < N] = -\ln \Psi - \sum_{c \in \mathcal{C}} \phi_c(\theta) \tag{74}$$

where Ψ is a normalizing constant and $\phi_c(\theta)$ is the potential associated to the c clique, $\mathcal{C}$ being the set of all cliques. As in [20], we define

$$\phi_c(\theta) = \begin{cases} -\zeta & \text{if all } \theta(i,j) \text{ in the clique are equal} \\ \zeta & \text{otherwise} \end{cases} \tag{75}$$

In what follows, we shall only consider 2-element cliques, with the same ζ parameters for all, *i.e.*, no orientation is preferential. The maximum *a posteriori* principle leads to the following minimization

$$-\ln Pr[\Theta = \theta, D = d] = -\ln Pr[\Theta = \theta] - \ln p(d|\Theta = \theta) \tag{76}$$

We have simplified the notation, but the above probabilities are to be taken for the whole picture. The probability density function linked to the observed inter-picture differences is

obtained by supposing that the noise is spatially independent. We suppose that the inter-picture difference is a Laplacian random variable (52). The Gaussian case is handled in the same manner. We can write

$$\ln p(d|\Theta=\theta) = \sum_{(i,j)\in S_0}\left(\ln\frac{\lambda_0}{2} - \lambda_0|d(i,j)|\right) + \sum_{(i,j)\in S_1}\left(\ln\frac{\lambda_1}{2} - \lambda_1|d(i,j)|\right) \tag{77}$$

where S_0 is the set of unchanged points and S_1 is the set of changed points. Thus can we express the quantity to be minimized in (3.76). Minimization can be achieved by applying a stochastic algorithm that uses a Gibbs sampler and the simulated annealing method [29]. This method can lead to the global minimum, but only to the expense of gigantic complexity or computation time. For the application in question, it would be preferable to apply deterministic algorithms, which provide a sub-optimal solution but are far less complex than the stochastic algorithm. We shall present further two such methods, as proposed by Derin and Elliot [20] and Besag [7].

Derin and Elliot [20] propose using Viterbi algorithm for the minimization of the formula (3.76) quantity. To achieve this, they write it in the recursive form, column by column,

$$\begin{array}{rcl} \Lambda_0 & = & \ln\Psi + MN\ln 2 \\ \Lambda_j & = & \Lambda_{j-1} + \sum_{c\in C^{j-1,j}}\phi_c(\theta) + \sum_{(i,j)\in S_0^j}(\lambda_0|d(i,j)| - \ln\lambda_0) \\ & & \sum_{(i,j)\in S_1^j}(\lambda_1|d(i,j)| - \ln\lambda_1); 0<j<N \end{array} \tag{78}$$

where $C^{j-1,j}$ is the set of cliques that have two points on column j, or on columns $j-1$ and j, and $S_n^j = \{(i,j) : \theta(i,j) = n; 0 \le i < M\}$. Finally for $j = N-1$ it will be

$$\Lambda_{N-1} = -\ln Pr[\Theta=\theta, D=d] \tag{79}$$

Using this recursion, it is possible to determine the solution by applying Viterbi algorithm. This solution will represent the global minimum. But the number of possible states for each column is 2^M, which may prove to be a prohibitive computation quantity. Therefore a sub-optimal solution will be enough, still using Viterbi algorithm, but applied in a sequential manner to a part of the column of height L. Then there will be 2^L possible states, but as many scans as there are lines in the picture will have to be performed.

Another method which takes the spatial relations expressed in the markovian field into account, is that proposed by Besag [7], and is called "iterated conditional modes" (ICM). This method generally provides a sub-optimal solution, for it uses local iterations. The method consists in starting from a given estimate, and hence update with all the locally available data. The quantity maximized at each point is the following conditional probability

$$Pr[\Theta(i,j)=\theta(i,j)|D(i,j)=d(i,j), \Theta(k,l)=\hat{\theta}(k,l); (k,l)\neq(i,j)] \tag{80}$$

This is equivalent to maximizing the following quantity

$$\ln p(d(i,j)|\theta(i,j)) + \ln Pr[\theta(i,j)|\hat{\theta}(k,l); (k,l)\in V(i,j)] \tag{81}$$

where $V(i,j)$ is the 8-point neighborhood of point (i,j). All the pixels can be updated at each step, but other implementations can be considered. As a rule, this algorithm is very fast, with few changes after the first six iterations. The approximation achieved, which corresponds to the local maximum of *a posteriori* probability $Pr\{\Theta(i,j)=\theta(i,j)|D(i,j)=d(i,j), 0\le i < M, 0\le j < N\}$ much depends on initialization. Initialization with a thresholding resulting from the maximum likelihood test should provide a good starting point for the "iterated conditional modes" algorithm.

In Figure 8 are given results of applying some of the above described methods on "Trevor White" sequence. For the case of decision at the point level, the model parameters have been estimated from the observed statistics of the inter-image difference ($\lambda_0 = 0.592$, $\lambda_1 = 0.037$, $P_0 = 0.8$ and $P_1 = 0.2$). The same values of parameters λ_0 and λ_1 have been used for the other two cases.

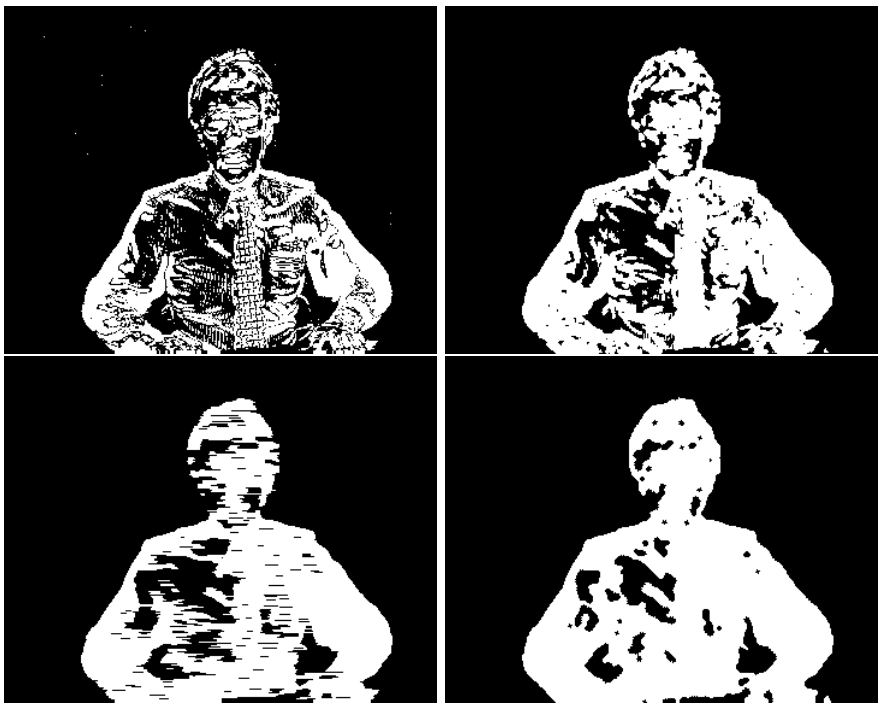

Figure 8: Results of change detection on "Trevor White" sequence. Top-left: independent decision point by point, one observation per point (Equation (3.56)). Top-right: independent decision point by point, nine (3×3) observations per point (Equation (3.64)). Bottom-left: decision at the line level. Bottom-right: decision at the 2-D level using the ICM algorithm

Before concluding this section on inter-frame change detection, let us remember that this detection can prove insufficient insofar as additional partitioning of the changed area can be sought, into a part belonging to the moving object, and a part of the background decovered as a result of motion. This would require either a finer modelization of the moving object and background, or using more than one pair of pictures [48].

3.2 Estimation of a 2-D apparent motion field

In this section we present methods to estimate apparent motion from spatio-temporal variations. The motion constraint equation, as it has been presented in Section 3.1, provides observation at all points of the picture. This measurement is not sufficient to estimate the apparent displacement vector, which has two components. This is local "unobservability" or ambiguity. Additional information has to be supplemented for the estimate to be achieved. This information generally is the coherence of the apparent displacement field. In practice, a form of spatial coherence is always provided. In a way, this additional information may be interpreted as a smoothing constraint within a regularization method. In a probabilistic expression of the estimation problem, coherence is expressed by an internal model, or state

model, of the two-dimensional field of apparent displacements. It is also possible to use a parametric model for the displacement field in a given region, or to perform the estimation in a transform domain.

In what follows, we present the main approaches and methods for displacement field estimation. Let us start with the simplest parametric model, which consists in supposing that in a given region displacement is translational. This method has been commonly used for picture compression, by matching rectangular blocks. We shall later present the recursive methods most studied and used for picture coding. Iterative methods have hardly been used for picture compression, and much used for picture analysis. Because of their methodological standing and their potential use for picture compression, we shall present them in quite some detail. In contrast, we shall be briefer with the methods that operate by spatio-temporal analysis within a transformed domain, since they are more cumbersome to implement and practically never used for picture coding. We shall finish with the estimation of the parametric models that are assumed to be valid within a given region. A question linked to these models, and more generally to those expressing displacement field coherence, is that of displacement field global ambiguity. The question is to determine the conditions, provided by the motion constraint equation and a displacement field model, under which there is a single solution to the displacement field within a given region. The answer to that question concludes this Section on estimation.

3.2.1 Translation of a 2-D region

Let us consider a region R, *i.e.*, a subset of connecting pixels. We suppose that this region moves according to a simple translation in two dimensions, which expresses the coherence of the velocity field. Let (u, v) be the displacement vector of the region. There are two possible ways of taking observations into account: from Equation (3.9), which is linear in relation to the unknown values, or from Equation (3.11).

i) Linear regression

The resolution of measurement Equations (3.9) over all the region pixels, is achieved by a least squares method, which provides the following solution

$$(\hat{u}, \hat{v}) = \underset{u,\, v}{\arg\min} \sum_{(i,j)\in R} (I_x u + I_y v + I_t)^2 \tag{82}$$

where spatial and temporal gradients are computed digitally on the set of pixels $(i, j) \in R$. Most often the time gradient is merely the inter-picture difference. Spatial gradients are computed on the preceding picture. The solution was provided by Cafforio and Rocca [17], as follows

$$\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \frac{1}{a_{11}a_{22} - a_{12}^2} \begin{bmatrix} a_{22} & -a_{12} \\ -a_{12} & a_{11} \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \tag{83}$$

with

$$\left.\begin{array}{rcl} a_{11} & = & \sum_{(i,j)\in R} I_x^2(i,j;k-1) \\ a_{22} & = & \sum_{(i,j)\in R} I_y^2(i,j;k-1) \\ a_{12} & = & \sum_{(i,j)\in R} I_x(i,j;k-1) I_y(i,j;k-1) \\ b_1 & = & \sum_{(i,j)\in R} I_x(i,j;k-1)(I(i,j;k) - I(i,j;k-1)) \\ b_2 & = & \sum_{(i,j)\in R} I_y(i,j;k-1)(I(i,j;k) - I(i,j;k-1)) \end{array}\right\} \tag{84}$$

Provided of course that $a_{11}a_{22} \neq a_{12}^2$. If a_{12} is negligible in comparison with the other parameters, the following simplification is obtained

$$\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \end{bmatrix} \tag{85}$$

This method permits measuring small displacements. Under the same translation hypothesis, but using Equation (3.11), a method, given hereafter, can be obtained to estimate greater displacements.

ii) Block matching

The block matching method is generally applied to a rectangular region whose size has been fixed. In addition, a search area is defined for the displacement vector. Be it

$$|u| \leq \Delta_h \quad \text{and} \quad |v| \leq \Delta_v \tag{86}$$

As a rule, the acceptable displacement vectors are multiple integers of the spatial sampling period, or of a half of this period. If the size of the block is K lines and L columns, the size of the search area, in the case of integer displacement, is $(K + 2\Delta_v)(L + 2\Delta_h)$. The number of possible displacement values, in the same case, is $(2\Delta_h+1)(2\Delta_v+1)$. For instance, for a vertically and horizontally maximal displacement of 7 pixels, the number of possible displacements is 225. The search area being defined, a criterion to be optimized has to be found. The criteria most frequently used are: the average absolute value displaced frame difference

$$(\hat{u}, \hat{v}) = \underset{\substack{(u,v) \in \mathbb{Z}^2 \\ |u| \leq \Delta_h, |v| \leq \Delta_v}}{\arg\min} \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} |I_{DFD}(i, j; u, v)| \tag{87}$$

the average squared displaced frame difference

$$(\hat{u}, \hat{v}) = \underset{\substack{(u,v) \in \mathbb{Z}^2 \\ |u| \leq \Delta_h, |v| \leq \Delta_v}}{\arg\min} \sum_{i=0}^{K-1} \sum_{j=0}^{L-1} (I_{DFD}(i, j; u, v))^2 \tag{88}$$

and the normalized average squared displaced frame difference

$$(\hat{u}, \hat{v}) = \underset{\substack{(u,v) \in \mathbb{Z}^2 \\ |u| \leq \Delta_h, |v| \leq \Delta_v}}{\arg\min} \frac{\sum_{i=0}^{K-1} \sum_{j=0}^{L-1} (I_{DFD}(i, j; u, v))^2}{\sum_{i=0}^{K-1} \sum_{j=0}^{L-1} I^2(i, j, k)} \tag{89}$$

where $I_{DFD}(i, j; u, v) = I(i, j; k) - I(i-u, j-v; k-1)$ is the displaced frame difference. Since the first criterion is less costly, it is used most often, knowing the weakness of the influence of these criteria on the displacement vector estimation.

Figure 9 shows results of applying block matching method on "Trevor White" sequence. The criterion of average absolute value of the displaced frame difference is used. The maximum horizontal or vertical displacement is 15.5 pixels, with an accuracy of a half pixel for the estimated displacement.

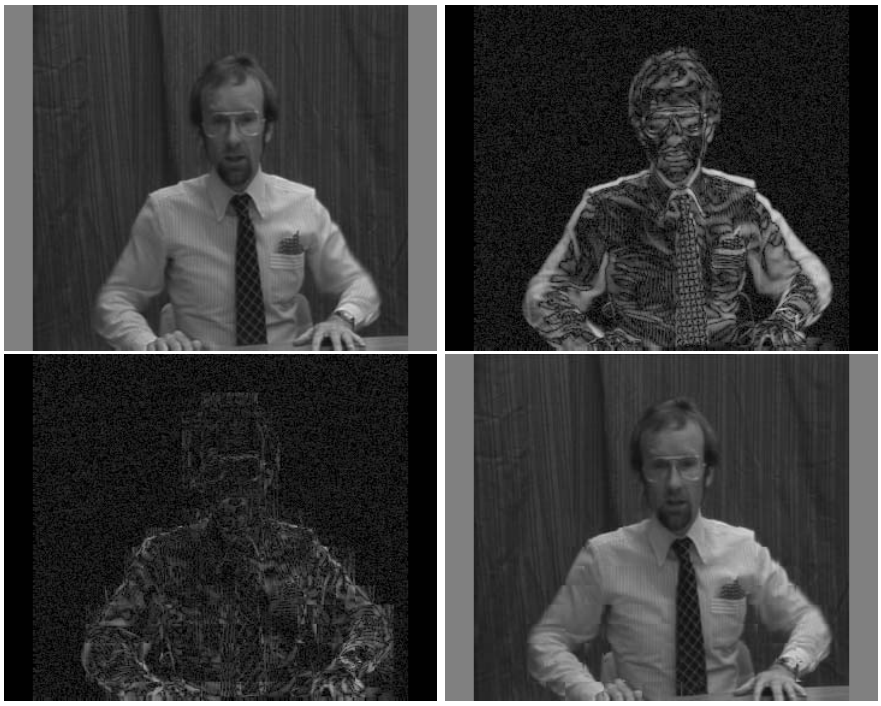

Figure 9: Results of block matching displacement estimation on "Trevor White" sequence. Top-left: the first frame. Top-right: the frame difference (amplified), MSE = 299.9. Bottom-left: the displaced frame difference (amplified), MSE = 30.9. Bottom-right: the predicted frame.

Although the operations necessary to estimate displacement are simple, real time implementation requires a very large computing capacity. For example, consider applying block matching to CIF standard format for videophone (Chapter 9) with a 30 frame/second rate. Supposing that maximal displacement is 7 horizontally and vertically, then the total number of operations per second is 684 288 000, where an operation comprises a substraction, an evaluation of the absolute value, and an accumulation. Several VLSI designs have been proposed [45] [77] [83], which meet real time computation requirements.

Also, several algorithms have been proposed, for fast search of the minimum, which are at least interesting for a computer implementation. The most common ones are presented hereafter.

a) 2-D logarithmic search [41]

This algorithm is based on the hypothesis that the criterium used for minimization is convex, and therefore that it increases in a monotonous manner with distance, in relation to the displacement which gives the minimal value. This represents an extension to the two dimensions of the one-dimensional logarithmic search. At each step, the search is done only at the five positions that include a middle point and four points in the two main directions, horizontal and vertical. In the first step, the four points are in the middle, between the center of the area and its border. If the optimum is at the middle point, among the five points, then the search area is decreased by half. This procedure continues until a 3×3 search area is obtained. In this last step, all nine positions are tested in order to determine the position of the minimum.

b) Increasing accuracy search [44]

This algorithm also is based on the criterion convexity hypothesis. In the first step, are tested displacement vectors over the search area with little accuracy. After determination of the displacement vector that minimizes the criterion among the nine possible, the process is pursued with increasing accuracy, until a one-pixel accuracy is obtained.

c) Conjugate direction search [68]

Starting from a null displacement, the minimum is first sought in one direction, the horizontal, for instance. Once the position of the minimum has been determined, the same search is carried out in the other direction, the vertical in our example. Once this search is complete, the same procedure is carried out in the direction determined by the point of origin and the last position of the minimum. A variant of this algorithm consists in skipping this last step, thus restricting the searching in the two main directions separately.

Figure 10 illustrates the three methods for fast search of the minimum for maximal (6,6) and a given (5,5) displacements.

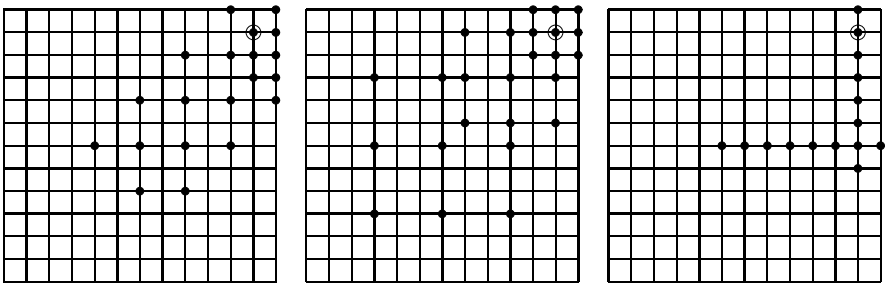

Figure 10: Minimum search for block matching. a) 2-D logarithmic; b) increasing accuracy; c) conjugate directions (simplified), for a displacement vector (5,5)

The following table [56] gives the maximal number of points tested by each of these fast algorithms, and the maximal number of steps necessary in the case of a 6-pixel maximal displacement, which in total would result in 169 points to be tested, exhaustively.

Algorithm	Max. number of points	Max. number of steps
2-D logarithmic	21	7
Increasing accuracy	25	3
Conjugate directions (simplified)	15	12

Table 1: Comparison of fast search algorithms for block matching

Santillana Rivero [66] compared the three algorithms of fast search of the minimum, and he concluded in favour of increased accuracy search, because with results comparable to those of 2-D logarithmic search, it can be carried out in fewer steps. Furthermore, this number of steps does not depend on the position of the minimum, but only on the maximal displacement. It is equal to $\lfloor \log_2(\max(\Delta_h, \Delta_v)) \rfloor + 1$.

Block matching algorithms provide only an integer accuracy on displacement. To obtain a sub-pixel accuracy, it is possible to use the same approach as that presented at the beginning

of this section, *i.e.*, the linear regression method, immediately after block matching. Let (u_m, v_m) be the displacement vector estimated using the block matching algorithm. The solution is expressed with a formula similar to (3.83)

$$\begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix} = \begin{bmatrix} \hat{u}_m \\ \hat{v}_m \end{bmatrix} + \frac{1}{a_{11}a_{22} - a_{12}^2} \begin{bmatrix} a_{22}b_1 - a_{12}b_2 \\ -a_{12}b_1 + a_{11}b_2 \end{bmatrix} \tag{90}$$

except that in the calculation of $\{a_{i,j}\}$ and $\{b_i\}$, the spatial gradient at the displaced point is used:

$$I_x(i - \hat{u}_m, j - \hat{v}_m; k-1) \text{ and } I_y(i - \hat{u}_m, j - \hat{v}_m; k-1)$$

and instead of the frame difference, that of the displaced frame is used

$$I(i, j; k) - I(i - \hat{u}_m, j - \hat{v}_m; k-1)$$

The larger the block, the more effective the block matching estimation is. But the larger the block, the lesser plausible is the hypothesis of an identical translation for all the pixels in the block. Confronted with such conflicting requirements, Bierling and Thoma [9] proposed a hierarchical approach for estimation. In the first step, a large block is considered, together with an important displacement. Later, the size of the block and the maximal displacement are reduced, and another search is carried out from the results of the preceding step. Bierling and Thoma [9] finally carried out three steps like these to obtain displacement as the sum of three displacement vectors. To limit computing complexity, it has been proposed to filter and sub-sample the picture for the first steps; this is done in such a way that the number of points in each block is practically constant, while covering a different part of the picture from one step to the other. This method relates to those of hierarchical or pyramidal estimation of motion, which are developed in Chapter 6.

3.2.2 Pixel-recursive algorithms

Point-by-point recursive algorithmic is also widely used for displacement estimation in picture coding. In this approach, an initial estimation is made for a given point, by prediction (so-called *a priori* estimation). This prediction expresses the spatial coherence of the displacement field. Prediction is generally made from the preceding point of the same line, but other prediction patterns are possible. A correction is later carried out according to the displaced frame difference, as computed by *a priori* estimation. This correction gives an estimate that can be called *a posteriori*. Knowing that the measurement equation is non-linear (3.11), it may be interesting to perform several local iterations of the correction procedure, to improve the *a posteriori* estimate. To present these recursive algorithms of prediction/correction, let us choose the theoretical framework of internal modelization, or state variable modelization, of the displacement field, and of the resultant optimal estimate, namely Kalman filtering.

The measurement equation is given in (3.11). We shall later write it in a form more suitable for solving the estimation problem.

$$I(i, j; k) = I(i - u(i, j), j - v(i, j); k-1) + e(i, j) + n(i, j) \tag{91}$$

where $(u(i, j), v(i, j))$ is the displacement vector, $e(i, j)$ is the predictable part of the process that expresses the difference between the two-dimensional displacement field and the optical flow (cf. Section 3.1 and Equation (3.8)), and $n(i, j)$ is an unpredictable white noise. We

obliterate the displacement vector, the $e(i,j)$ and the noise $n(i,j)$ dependence on the time index, because we only use the spatial coherence in the estimation, and the estimation will be carried out using only the inter-frame difference. The state vector which characterizes the displacement field associated with intensity variations is given by

$$\xi(i,j) = \begin{bmatrix} u(i,j) \\ v(i,j) \\ e(i,j) \end{bmatrix} \tag{92}$$

In what follows, we suppose that the state vector is only recursive in one direction, and therefore the direction of picture scanning is chosen accordingly. As it as been mentioned, several choices are possible [22], [23]. This gives the following state equation

$$\xi(i,j) = \Phi(i,j)\xi(i-1,j) + q(i,j) \tag{93}$$

where $q(i,j)$ is a model noise vector. Let us first suppose that the state vector is stationary, in the sense that the transition matrix $\Phi(i,j)$ is invariant for the whole picture, as are the statistical characteristics of $q(i,j)$. Later, we shall study the case where these characteristics depend on position (i,j). With regard to the rupture-type non-stationarities, these are dealt with separately in Section 3.3.3 on displacement field segmentation with a discontinuity detector. Back to the stationarity hypothesis. We suppose that

$$\Phi(i,j) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \rho \end{bmatrix} = \Phi, 0 \leq \rho \leq 1 \tag{94}$$

and that $q(i,j)$ is zero-mean with

$$E\{q(i,j)q^T(l,m)\} = \begin{bmatrix} \sigma_v^2 & 0 & 0 \\ 0 & \sigma_v^2 & 0 \\ 0 & 0 & \sigma_e^2 \end{bmatrix} \delta(i,l)\delta(j,m) = Q\delta(i,l)\delta(j,m) \tag{95}$$

With this model and a non-linear measurement equation, it is necessary to use the extended Kalman filter to estimate the state vector. The solution was given by Jazwinski ([43], p. 278). The *a priori* estimate of the state vector at point (i,j) is given by

$$\begin{array}{rcll} \hat{\xi}^0(i,j) & = & \Phi\hat{\xi}(i-1,j), & i>0 \\ \hat{\xi}^0(0,j) & = & \Phi\hat{\xi}(N-1,j-1), & j>0 \\ \hat{\xi}^0(0,0) & = & 0 & \end{array} \tag{96}$$

This recursion, for going from one line to the next, may be advantageous in case of an homogeneous motion, where the state vector is stationary with regard to both directions. The case of a rupture may also be taken into account with the discontinuity detector which will be presented in Section 3.3.3. The covariance matrix of the *a priori* estimate is given by

$$\begin{array}{rcll} P^0(i,j) & = & \Phi P(i-1,j)\Phi^T + Q, & i>0 \\ P^0(0,j) & = & \Phi P(N-1,j-1)\Phi^T + Q, & j>0 \\ P^0(0,0) & = & \alpha Q & \end{array} \tag{97}$$

with $\alpha \gg 1$.

Taking the intensity observation at point (i,j) into account, we obtain the *a posteriori* estimate as follows :

$$\hat{\xi}(i,j) = \hat{\xi}^0(i,j) + K(i,j)\left(I(i,j;k) - I(i-\hat{u}^0, j-\hat{v}^0; k-1) - \hat{e}^0\right) \tag{98}$$

where $K(i,j)$ is the Kalman gain value produced by

$$K(i,j) = \frac{P^0(i,j)G(i,j)}{G^T(i,j)P^0(i,j)G(i,j) + R} \tag{99}$$

where the $G(i,j)$ vector is the gradient of observation in relation to the state vector components

$$G(i,j) = -\begin{bmatrix} I_x(i-\hat{u}^0, j-\hat{v}^0; k-1) \\ I_y(i-\hat{u}^0, j-\hat{v}^0; k-1) \\ -1 \end{bmatrix} \tag{100}$$

and R is the variance of observation noise $n(i,j)$.

Finally, the covariance matrix of the *a posteriori* estimate is computed as

$$\begin{aligned} P(i,j) = &(\mathbf{I}_3 - K(i,j)G^T(i,j))P^0(i,j)(\mathbb{I}_3 - K(i,j)G^T(i,j))^T \\ &+ RK(i,j)K^T(i,j) \end{aligned} \tag{101}$$

Since the displacement vector estimation has a sub-pixel accuracy, the intensity involved in formula (3.98), together with the spatial gradients of (3.100) have to be interpolated. Most often two-dimensional linear interpolation from the four nearest points is used. We give its expression for intensity, its transposition to the spatial gradients being obvious, by masking the time index

$$\begin{aligned} I(i-u, j-v) = &(1-d_v)(1-d_u)I(i-\lfloor u \rfloor, j-\lfloor v \rfloor) \\ &+(1-d_v)d_u I(i-\lfloor u \rfloor - 1, j-\lfloor v \rfloor) + d_v(1-d_u)I(i-\lfloor u \rfloor, j-\lfloor v \rfloor - 1) \\ &+d_v d_u I(i-\lfloor u \rfloor - 1, j-\lfloor v \rfloor - 1) \end{aligned} \tag{102}$$

with $u = \lfloor u \rfloor + du$, $v = \lfloor v \rfloor + dv$, and $0 \leq du \leq 1$, $0 \leq dv \leq 1$.

From this structure, it is possible, under certain hypotheses, to obtain other estimators which in general are more simple with regard to computing complexity. Let us first suppose that $\rho = 0$ and $\sigma_e^2 = 0$, or equally that $e(i,j) = 0$. Then the state vector is reduced to the displacement vector, and therefore its dimension is reduced to 2, or

$$\xi(i,j) = \begin{bmatrix} u(i,j) \\ v(i,j) \end{bmatrix} \tag{103}$$

with the state equation

$$\xi(i,j) = \xi(i-1,j) + q(i,j) \tag{104}$$

$q(i,j)$ being a two-dimensional vector, with the same type of hypotheses as above. The same type of filter is obtained for the estimation with

$$G(i,j) = -\begin{bmatrix} I_x(i-\hat{u}^0, j-\hat{v}^0; k-1) \\ I_y(i-\hat{u}^0, j-\hat{v}^0; k-1) \end{bmatrix} \tag{105}$$

This estimation algorithm was studied by Stuller and Krishnamurthy [69] and by Mijiyawa [53].

Let us now hypothesize that the two-dimensional vector $q(i,j)$ is not spatially stationary, but that the covariance matrix of this vector is given by

$$Q(i,j) = \frac{\sigma_v^2 G(i,j)G^T(i,j)}{\mu + G^T(i,j)G(i,j)}, \quad \mu = \frac{R}{\sigma_v^2} \tag{106}$$

This hypothesis means that the normal component variance, in the direction of the gradient, of the state noise vector is given by

$$E\{\frac{(G^T(i,j)q(i,j))^2}{||G(i,j)||^2}\} = \frac{\sigma_v^2||G(i,j)||^2}{\mu + ||G(i,j)||^2} \tag{107}$$

Therefore, this variance increases with the spatial gradient, and conversely in small gradient areas the displacement vector does not vary much. With regard to the tangential component of the velocity vector, this does not vary from one point to the other under the above hypothesis; in contrast, the direction of this component obviously may vary.

We observe that with this expression of the covariance matrix of the state noise (3.106) the solution of matrix equations involving the covariance matrices of *a priori* and *a posteriori* estimates give the following solution

$$P^0(i,j) = \sigma_v^2 \mathbb{I}_2 \tag{108}$$

and

$$P(i,j) = \sigma_v^2 \left(\mathbf{I}_2 - \frac{G(i,j)G^T(i,j)}{\mu + G^T(i,j)G(i,j)} \right) \tag{109}$$

Thus in this context, σ_v^2 can be interpreted as the variance of the *a priori* estimate for each of the velocity vector components. In addition, the gain for the *a posteriori* estimation is given by

$$K(i,j) = \frac{G(i,j)}{\mu + ||G(i,j)||^2} \tag{110}$$

The algorithm thus obtained is that initially and differently introduced by Cafforio and Rocca [18]. The demonstration that this algorithm is obtained in a state model expression was provided by Pesquet [61]. Figure 11 shows results produced by the Cafforio-Rocca algorithm on "Trevor White" sequence.

It is also possible to base the *a posteriori* estimation on more than one observation, taking into account causally anterior points of the vicinity of (i,j). Let p be the number of observations, then we obtain a $p \times 1$ size observation vector, whose covariance matrix is supposed diagonal and equal to $R\mathbf{I}_p$. Spatial gradients are given in a matrix $G(i,j)$ $2 \times p$ in size. Supposing, as in the case of a single observation, that

$$Q(i,j) = \sigma_v^2 G(i,j)(\mu \mathbf{I}_p + G^T(i,j)G(i,j))^{-1} G^T(i,j) \tag{111}$$

with $\mu = \frac{R}{\sigma_v^2}$, then the gain takes the following form (Appendix 3F)

$$K(i,j) = G(i,j)(\mu \mathbb{I}_p + G^T(i,j)G(i,j))^{-1} \tag{112}$$

which is a $2 \times p$ size matrix. We see that thus we obtained the algorithm of Biemond, Looijenga and Boekee [8] (Appendix 3F).

Going back to a single observation, and supposing furthermore that the variance of the measurement noise varies spatially according to the spatial gradient

$$R(i,j) = \sigma_v^2 \|G(i,j)\|^2 \tag{113}$$

Thus the measurement noise is assumed to be stronger as the spatial gradient is greater. The following gain is then obtained from (3.110)

$$K(i,j) = \frac{G(i,j)}{2\|G(i,j)\|^2} \ , \quad \|G(i,j)\| > 0 \tag{114}$$

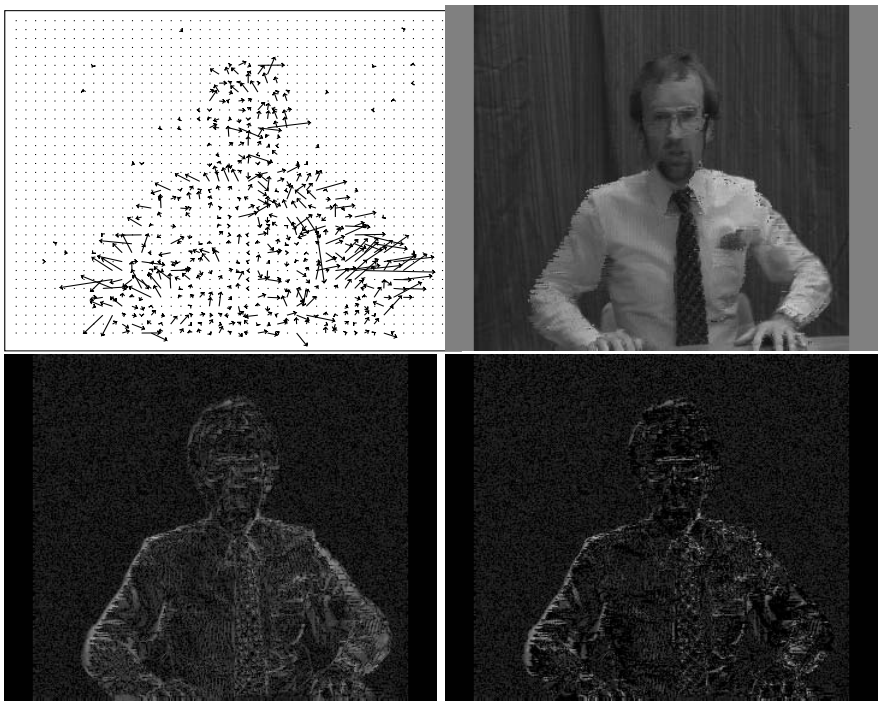

Figure 11: Results of recursive displacement estimation on "Trevor White" sequence using the Cafforio-Rocca algorithm. Top-left: the resulting 2-D velocity field (multiplied by 3). Top-right: the predicted frame using the *a posteriori* estimated velocity field. Bottom-left: the *a priori* displaced frame difference (amplified), MSE = 38.89. Bottom-right: the *a posteriori* displaced frame difference (amplified), MSE = 28.85.

This corresponds to the algorithm of Walker and Rao [78].

This algorithm, as well as that of Cafforio and Rocca (3.110), have the advantage of being less complex than Kalman filter, whose gain is given in (3.99), since there are no matrix equations to be resolved, *i.e.*, there are no recursions to be performed on the covariance matrices $P^0(i,j)$ and $P(i,j)$.

To simplify still further, let us suppose that R is spatially invariant, and furthermore

$$R \gg \sigma_v^2 \|G(i,j)\|^2 \tag{115}$$

From (3.110) we obtain

$$K(i,j) = \epsilon G(i,j) \tag{116}$$

with $\epsilon = \frac{\sigma_v^2}{R}$. This is the algorithm proposed by Netravali and Robbins [58]. Seen from this presenting angle, the adaptation step ϵ must meet the following inequality

$$0 < \epsilon < \frac{1}{\max\limits_{0 \le i < N, 0 \le j < M} \|G(i,j)\|^2} \tag{117}$$

It is also possible to view this algorithm from a different angle, with less constraints on the adaptation step size. This framework was chosen by Netravali and Robbins to introduce

their algorithm, *i.e.*, the stochastic gradient. In this precise context, it has been proven [50] that convergence is ensured, if

$$0 < \epsilon < \frac{2}{E\{\|G(i,j)\|^2\}} \tag{118}$$

This condition for convergence was also given in [58], where mathematical expectation is replaced by an empirical mean for the whole moving region. Analysis of this algorithm is effected under the hypothesis that the displacement vector is constant, or that motion is a mere translation along the picture plane. The above-mentioned analysis is carried out on a linear observation model (3.9). Moorhead, Rajala and Cook [54], taking the non-linear observation model into account, obtained a convergence condition closer to the real model. In any case, a trade-off has to be found between convergence speed and permanent state fluctuations. Indeed these two objectives are contradictory. The greater ϵ is, satisfying (3.118), the shorter the transitory, but the greater the estimation fluctuations, which are proportional to the adaptation step size [50]. A suitable compromise is given by the following value

$$\epsilon = \frac{1}{E\{\|G(i,j)\|^2\}} \tag{119}$$

With regard to the algorithm, whose gain in given in (3.110), it may be viewed within the normalized stochastic gradient. It has been shown [10] that under certain, not very strict conditions, the algorithm converges towards the real displacement vector, when the vector is constant. These conditions are given in Appendix 3D. These results can also be applied, in practice, to the algorithm of Walker and Rao (3.114).

For all the pel-recursive algorithms presented in this section, and because of the non-linearity of the measurement equation, it may be useful to perform several local iterations at each point in order to improve the *a posteriori* estimate [4] [73] [78].

We end this section by mentioning the existence of VLSI circuits developed for motion pel-recursive estimators [15], [27].

3.2.3 Iterative algorithms

This category of algorithms encompasses the methods which update velocity vectors in parallel over the whole frame, convergence being later obtained after a number of iterations. This type of algorithm is practically never used for coding; but it could be, and it raises great interest in picture analysis, which justifies its presence in this book. Within this framework, the motion constraint equation is used in the linear form (3.9). A regularization method is used to minimize jointly the error from constraint-linked measurement of motion on the one hand, and the error from a model that expresses the coherence of the velocity field on the other.

Horn and Schunck [37] proposed using a first order smoothing constraint, *i.e.*, to limit to the smallest value the gradient module of two components of optical flow. We first give the criterion to be minimized in continuous coordinates for a clearer illustration, and the following development is given in discrete coordinates

$$\iint ((I_x u + I_y v + I_t)^2 + \lambda(u_x^2 + u_y^2 + v_x^2 + v_y^2))dxdy$$

where u_x, u_y, v_x and v_y are the partial first derivatives of the two optical flow components, in relation to the (x, y) coordinates. In the discrete form, it is necessary to minimize the

following quantity over a certain region, or the whole frame

$$\begin{aligned}\sum\sum(I_x(i,j;k-1)u(i,j)+I_y(i,j;k-1)v(i,j)+I(i,j;k)-I(i,j;k-1))^2\\ +\lambda\sum\sum((u(i,j)-u(i-1,j))^2+((u(i,j)-u(i,j-1))^2\\ +\lambda\sum\sum((v(i,j)-v(i-1,j))^2+((v(i,j)-v(i,j-1))^2)\end{aligned} \tag{120}$$

Minimization of this quantity provides the following system of linear equations

$$\left.\begin{array}{rcl}\lambda(u-\bar{u})+I_x(I_xu+I_yv+I_t) & = & 0\\ \lambda(v-\bar{v})+I_y(I_xu+I_yv+I_t) & = & 0\end{array}\right\} \tag{121}$$

where all indices have been omitted to simplify notation. In Equations (121), $\bar{u}$ (resp. $\bar{v}$) is a linear interpolation of the u (resp. v) field, as given for example by

$$\bar{u}(i,j)=0.25(u(i-1,j)+u(i+1,j)+u(i,j-1)+u(i,j+1)) \tag{122}$$

The system of Equations (121) may also be written

$$\left.\begin{array}{l}(\lambda+I_x^2)(u-\bar{u})+I_xI_y(v-\bar{v})=-I_x(I_x\bar{u}+I_y\bar{v}+I_t)\\ I_xI_y(u-\bar{u})+(\lambda+I_y^2)(v-\bar{v})=-I_y(I_x\bar{u}+I_y\bar{v}+I_t)\end{array}\right\} \tag{123}$$

and finally

$$\begin{bmatrix}u\\v\end{bmatrix}=\begin{bmatrix}\bar{u}\\\bar{v}\end{bmatrix}-\frac{I_x\bar{u}+I_y\bar{v}+I_t}{\lambda+I_x^2+I_y^2}\begin{bmatrix}I_x\\I_y\end{bmatrix} \tag{124}$$

In this form, the system of linear equations can be resolved using the Gauss-Seidel method, which is very adapted due to the size of the system and the fact that the corresponding matrix is null outside of an area surrounding the diagonal. Iterations are carried out as follows

$$\begin{bmatrix}u\\v\end{bmatrix}^m=\begin{bmatrix}\bar{u}\\\bar{v}\end{bmatrix}^{m-1}-\frac{I_x\bar{u}^{m-1}+I_y\bar{v}^{m-1}+I_t}{\lambda+I_x^2+I_y^2}\begin{bmatrix}I_x\\I_y\end{bmatrix} \tag{125}$$

with m the iteration index. In [37] the numerical computing of the spatio-temporal derivatives is different, but it corresponds to velocity vectors not localized at the same points as the positions resulting from intensity sampling, either spatially or temporally. Figure 12 shows a subsampled version of the optical flow field produced by the Horn-Schunck algorithm on "Trevor White" sequence.

The smoothing constraint used in (3.120) corresponds to an essentially translational motion. If spatial relations (3.18) are used to express the coherence of the two-dimensional velocity field, then the smoothing constraint is modified, and therefore the resolution algorithm of the linear system is modified too. But the difference is limited to the computation of the field linear interpolation. In the case of a smoothing constraint derived from (3.18), it is [73]

$$\begin{aligned}\bar{u}(i,j)=\frac{1}{20}(8(u(i-1,j)+u(i+1,j)+u(i,j-1)+u(i,j+1))\\ -2(u(i-1,j-1)+u(i-1,j+1)+u(i+1,j-1)+u(i+1,j+1))\\ -(u(i-2,j)+u(i+2,j)+u(i,j-2)+u(i,j+2))\end{aligned} \tag{126}$$

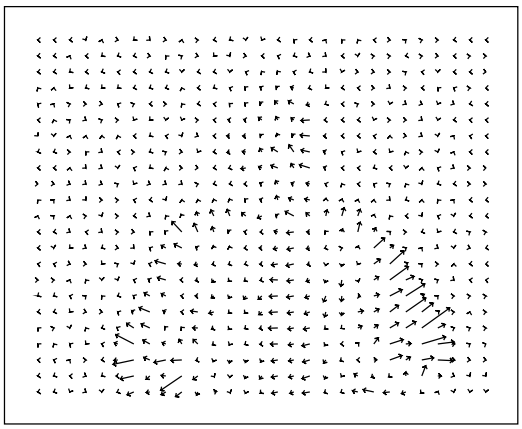

Figure 12: An optical flow field for the Horn-Schunck algorithm and the "Trevor White" sequence

Due to a vertically and horizontally different sample, it may be necessary to introduce different regularization factors into the two velocity field components. Let λ_1 and λ_2 be these two factors for u and v respectively. Equation (3.125) is modified as follows

$$\begin{bmatrix} u \\ v \end{bmatrix}^m = \begin{bmatrix} \bar{u} \\ \bar{v} \end{bmatrix}^{m-1} - \frac{I_x \bar{u}^{m-1} + I_y \bar{v}^{m-1} + I_t}{\lambda_1\lambda_2 + \lambda_2 I_x^2 + \lambda_1 I_y^2} \begin{bmatrix} \lambda_2 I_x \\ \lambda_1 I_y \end{bmatrix} \tag{127}$$

The smoothing constraint in (3.120) is perfectly isotropic in relation to the frame plane coordinates. Nagel [57] proposed a smoothing constraint oriented according to the spatial gradient of intensity. More precisely, only the perpendicular component to the spatial gradient has to vary slightly. This has to be related to the discussion on the covariance matrix of the state noise given in (3.106), in the recursive algorithm situation. We only give Nagel's smoothing constraint for one of the two components, since these two components are considered to be decoupled in the smoothing constraint,

$$\frac{(I_y^2+\gamma)u_x^2 - 2I_x I_y u_x u_y + (I_x^2+\gamma)u_y^2}{I_x^2+I_y^2+2\gamma}$$

A similar solution to (3.124) can be obtained with this smoothing constraint, and the same type of iterative algorithm can be used to resolve the resultant system of linear equations.

It is also possible, in an iterative method, to take into account the difference between the two-dimensional velocity field and the optical flow, as illustrated in (3.8) for example, and as a part of the recursive method using the observation equation (3.91). This approach was proposed by Cornelius and Kanade [19], by introducing an additional smoothing constraint on $e(i,j)$ in (3.91)

$$\mu \sum\sum ((e(i,j) - e(i-1,j))^2 + (e(i,j) - e(i,j-1))^2)$$

Thus are obtained three types of equations

$$\left.\begin{array}{c} \lambda(u-\bar{u}) + I_x(I_x u + I_y v + I_t - e) = 0 \\ \lambda(v-\bar{v}) + I_y(I_x u + I_y v + I_t - e) = 0 \\ \mu(e-\bar{e}) - (I_x u + I_y v + I_t - e) = 0 \end{array}\right\} \tag{128}$$

These equations can also be written as follows

$$\left.\begin{array}{rcl} (\lambda + I_x^2)(u - \bar{u}) + I_x I_y (v - \bar{v}) - I_x (e - \bar{e}) & = & -I_x \Delta I \\ I_x I_y (u - \bar{u}) + (\lambda + I_y^2)(v - \bar{v}) - I_y (e - \bar{e}) & = & -I_y \Delta I \\ -I_x (u - \bar{u}) - I_y (v - \bar{v}) + (\mu + 1)(e - \bar{e}) & = & \Delta I \end{array}\right\} \quad (129)$$

where $\bar{u}$ and $\bar{v}$ bear the same significance as before, $\bar{e}$ is also a linear interpolation of the same type and $\Delta I = I_x \bar{u} + I_y \bar{v} + I_t - \bar{e}$. The solution in relation to u, v and e leads to the following equations

$$\begin{bmatrix} u \\ v \\ e \end{bmatrix} = \begin{bmatrix} \bar{u} \\ \bar{v} \\ \bar{e} \end{bmatrix} - \frac{I_x \bar{u} + I_y \bar{v} + I_t - \bar{e}}{(\mu + 1)\lambda + \mu(I_x^2 + I_y^2)} \begin{bmatrix} \mu I_x \\ \mu I_y \\ -\lambda \end{bmatrix} \quad (130)$$

It is therefore possible to use the Gauss-Seidel algorithm for the resolution of this system of equations.

3.2.4 Analysis of spatio-temporal frequencies

Estimation methods have recently been proposed, which are based on spatio-temporal filters and which use several pictures. As a result, these methods are of little interest for picture sequence coding. Therefore they are presented only briefly. These methods are based on a fundamental feature of Fourier transformation in a spatio-temporal space, consisting in concentrating, within the frequential domain, all the energy on a single plane. For spatial frequencies (ω_x, ω_y) and temporal frequency ω_t, the following equation is obtained for optical flow

$$u\omega_x + v\omega_y + \omega_t = 0 \quad (131)$$

Thus for a spatio-temporal frequency triplet we obtain an optical flow equation, and therefore there is a local ambiguity in the frequency domain. Analysis over the whole domain permits estimating the optical flow (u, v). This presentation is more relevant to the principle of the approach, by supposing that the picture is stationary. In fact, the three-dimensional signal (2-D+time) is not stationary; it is therefore necessary to perform an analysis simultaneously spatio-temporal and frequential. Thus Fourier transformation is performed in a limited part of the 2-D space and time. Several spatiotemporal filters are used, which are of the bandpass type, and which practically cover the whole frequential domain. There are several alternatives to this spatio-temporal and frequential analysis : local Fourier analysis, Wigner transformation [40], and Gabor transformation [31].

3.2.5 Parametric model

In a way, this approach is the generalization of the estimation of a translational displacement presented first in this section. This time, the motion model may comprise more than two parameters. Two examples of possible parametering are given in formulas (3.15) and (3.17). Then we take formulation of (3.17). The same approach could be applied to parametering (3.15) or to any other parametering of the two-dimensional velocity field [35] [59] [60].

Parameter estimation is carried out on a region of the frame, as may result either from segmentation (Section 3.3.3) or from the division of the frame in blocks. The model (3.17) can be written more simply

$$\begin{array}{rcl} u & = & a_{10} + a_{11} x + a_{12} y \\ v & = & a_{20} + a_{21} x + a_{22} y \end{array} \quad (132)$$

This model was obtained in Section 3.1 as a result of a certain projection on the picture plane of the three-dimensional velocity field, but it may be useful in other situations, outside of the restrictive context that gave the Equations (3.17). The main point is that this model can be interpreted as a two-dimensional motion including at the same time translation, rotation and deformation. Contraction and dilation are examples of deformation.

Restricting ourselves to one region, the criterion to be minimized most often used is the quadratic mean of the displaced frame difference. Taking θ as the parametric vector

$$\theta = \begin{bmatrix} a_{10} & a_{11} & a_{12} & a_{20} & a_{21} & a_{22} \end{bmatrix}^T \tag{133}$$

We attempt to determine the following estimation

$$\hat{\theta} = \arg\min \sum_{(i,j)\in R} (I(i,j;k) - I(i-u, j-v; k-1))^2 \tag{134}$$

To do so, we present hereafter a gradient method [25] [59] [60]. The estimate of θ is obtained in an iterative manner

$$\hat{\theta}_m = \hat{\theta}_{m-1} - \mu \sum_{(i,j)\in R} \phi_{m-1}(i,j) \left(I(i,j;k) - \hat{I}(i,j;k) \right) \tag{135}$$

where $\hat{I}(i,j;k) = I(i - \hat{u}_{m-1}, j - \hat{v}_{m-1}; k-1)$, m is the iteration index, u_{m-1} and v_{m-1} are obtained using model (132), and $\phi_{m-1}(i,j)$ is a vector dependent on the spatial gradients of intensity. The generic form of this gradient vector is the following

$$\phi(i,j) = \begin{bmatrix} I_x(i-u, j-v; k-1) \\ iI_x(i-u, j-v; k-1) \\ jI_x(i-u, j-v; k-1) \\ I_y(i-u, j-v; k-1) \\ iI_y(i-u, j-v; k-1) \\ jI_y(i-u, j-v; k-1) \end{bmatrix} \tag{136}$$

The adaptation step size μ must satisfy the inequality $\mu < \frac{2}{\lambda_{\max}}$, where $\lambda_{\max}$ is the greatest eigenvalue of the matrix $\sum_{(i,j)\in R} \phi(i,j)\phi^T(i,j)$. This therefore is the steepest descent method.

Another approach is that of the normalized gradient type. This requires performing, between iterations, a linear approximation in relation to the parameters of the displaced frame difference

$$\begin{aligned} &I(i,j;k) - I(i-u, j-v; k-1) = \\ &\quad I(i,j;k) - I(i-u_{m-1}, j-v_{m-1}; k-1) + (\theta - \theta_{m-1})^T \phi_{m-1}(i,j) \end{aligned} \tag{137}$$

Under these conditions, minimization of the above criterion gives

$$\begin{aligned} \hat{\theta}_m = \hat{\theta}_{m-1} - &\left(\sum_{(i,j)\in R} \phi_{m-1}(i,j)\phi_{m-1}^T(i,j) \right)^{-1} \\ &\sum_{(i,j)\in R} \phi_{m-1}(i,j)(I(i,j;k) - I(i-\hat{u}_{m-1}, j-\hat{v}_{m-1}; k-1)) \end{aligned} \tag{138}$$

The iterative aspect of the algorithm is justified by the non-linearity of intensity measurements. Dugelay and Sanson [25] proposed a simplified version of (3.138), replacing the matrix

$\sum_{(i,j)\in R} \phi(i,j)\phi^T(i,j)$ by an identity matrix multiplied by the trace of the matrix that contains the gradients. This avoids inverting this matrix, and gives the following iteration

$$\hat{\theta}_m = \hat{\theta}_{m-1} - \frac{\sum\limits_{(i,j)\in R} \phi_{m-1}(i,j)(I(i,j;k) - \hat{I}(i,j;k))}{\sum\limits_{(i,j)\in R} \phi^T_{m-1}(i,j)\phi_{m-1}(i,j)} \tag{139}$$

3.2.6 Global ambiguity of the 2-D velocity field

In this section (3.3.2), we have raised the question of using various models of field coherence and different method of coherence exploitation, to estimate the two-dimensional velocity field. According to the model used, there may be an infinity of solutions compatible with the observations provided by the motion constraint equation. This may even be critical in the case where a division or a matrix inversion is carried out to resolve the estimation problem, as for instance in (3.83), (3.85), (3.138) and (3.139). This paragraph reviews the situations where a global ambiguity of the 2-D velocity field may appear in a region.

A global ambiguity of the 2-D velocity field is said to exist if, given a model for the field and an observation equation of the type

$$I_x u + I_y v = 0 \tag{140}$$

in a region, there are, beside the obvious solution $(u, v) = (0, 0)$, other non-null solutions. Hereafter we propose conditions for global ambiguity.

Proposition. If the intensity function can be written as $I(g(x, y))$, where $g(x, y) = 0$ is the equation of a curve on the plane that satisfies the Equation (3.42)

$$\frac{dy}{dx} = \frac{v}{u}$$

then there is an infinity of solutions for the whole set of Equations (3.140) in the region.

Demonstration. Equation (3.140) writes

$$I_g(g_x u + g_y v) = 0 \tag{141}$$

where I_g is the derivative of I in relation to function g. In the case where $I_g = 0$, it is obvious that any (u, v) satisfies the Equation (3.141). If $I_g \neq 0$, this equation is equivalent to $g_x u + g_y v = 0$. Knowing that the vector $\begin{bmatrix} g_x & g_y \end{bmatrix}^T$ is the normal vector to the curve $g(x, y) = 0$, and that this vector is colinear with $\begin{bmatrix} dy & -dx \end{bmatrix}^T$, we obtain the Equation (3.42) of the above proposition.

Let us consider the consequence of this result on two models widely used in this section. First, the simple translation model. The solution to Equation (3.42) is a straight line, which means that the intensity function takes the form of $I(\alpha x + \beta y + \gamma)$. As to the linear model (3.132), the solution to (3.42) is given in Section 3.2.3.

3.3 Segmentation of the two-dimensional velocity field

Segmentation of the two-dimensional velocity field is an important step of image sequence analysis, especially to determine the 3-D motion characteristics of the objects present in the scene. There are two types of spatio-temporal boundaries: occlusion boundaries between objects, and boundaries between different regions of the same object, *i.e.*, those points on the object surface that are characterized by strong 3-D curvature.

There are two approaches to be used in the determination of these boundaries. The first one operates after estimation of a dense velocity field. The second approach poses the estimation and segmentation problems jointly. Indeed, all estimation methods use the spatial coherence of the 2-D velocity field to get away from the indetermination of the field from measurements alone. It is necessary to know the boundaries between spatio-temporal regions to apply a predictor or an interpolator of the velocity field, and therefore to correctly estimate that field. Also, field estimation is necessary for segmentation. This problem relates to that of the "chicken and egg". To break that vicious circle, a combined method of estimation/segmentation is used. Different methods pertaining to this strategy are presented further; but let us linger on the first approach.

Thompson, Mutch and Berzins [71] considered the case of occlusion boundaries. In that case, the module and/or the direction of the displacement vector contains discontinuities. These are detected with a Laplacian of Gaussian operator applied to the two velocity field components.

Adiv [2] proposed a three-step method to carry out segmentation of the velocity field. In the first step, a grouping is performed with Hough transformation according to the linear model (3.132). In the second step, the resultant parts of the first step are merged using the 8-parameter model (3.41). In the ultimate step, the points that were not grouped in the first step are merged to the region with the greatest coherence with the model (3.41).

3.3.1 Pel-recursive algorithmic

In pel-recursive algorithmic, ruptures are detected, rather than proper segmentation is performed. All pel-recursive algorithms include two phases : spatial prediction and correction according to the displaced frame difference. Discontinuity detection fits in between the two. The hypothesis of velocity field spatial coherence is verified as valid or not. To do so, the predicted velocity field is tested against the hypothesis of a null displacement. Note H_0 being the spatial coherence hypothesis and H_1 the hypothesis of discontinuity, with *a priori* probabilities of P_0 and P_1, respectively. We suppose that the displaced frame difference of hypothesis H_0 is a random zero-mean variable with a variance of σ^2, which is supposed known to determine the test. Under hypothesis H_1, we suppose that the inter-frame difference is also a zero-mean random variable with the same variance.

The test is obtained by using the criterion of *a posteriori* probability maximization. Two cases are to be distinguished according to the probability law of the inter-frame difference. If this probability law is Gaussian, a discontinuity is detected, if

$$(I(i,j;k) - \hat{I}(i,j;k))^2 - (I(i,j;k) - I(i,j;k-1))^2 > 2\sigma^2 \ln \frac{P_0}{P_1} \tag{142}$$

where $\hat{I}(i,j;k) = I(i - \hat{u}^0, j - \hat{v}^0; k-1) - \hat{e}^0$. If the law is Laplacian, the following test is obtained for the same decision

$$|I(i,j;k) - \hat{I}(i,j;k)| - |I(i,j;k) - I(i,j;k-1)| > \frac{\sigma}{\sqrt{2}} \ln \frac{P_0}{P_1} \tag{143}$$

These tests can apply when the algorithm also estimates the variable $e(i,j)$ as described by (3.96)-(3.101). If the deviation $e(i,j)$ has not been estimated, the above tests are modified accordingly.

In principle, and for the above tests, detection of a discontinuity results in initialization of the displacement vector: $\hat{u}^0 = \hat{v}^0 = 0$. Labit and Benveniste [47] proposed, in case of prior estimation of the contour motion, and in case of discontinuity detection, to reinitialize

the displacement *a priori* estimate according to the available estimate of displacement on the contour. This reinitialization can be carried out systematically, or after a test that will determine to which part the contour belongs.

3.3.2 Iterative algorithmic

Iterative algorithms as they were presented in paragraph 3.3.2 smooth the velocity field across boundaries, between different spatio-temporal regions. Thus the velocity field of a moving object may be propagated on its fixed surrounding background. A method to avoid this type of deformation is to detect the moving areas or the inter-frame changes, and to apply the iterative algorithm only to that area. But this approach does not fit any type of discontinuity. The best approach consists in integrating explicitly in the optimization criterion the discontinuities of the velocity field [32] [33] [39]. The criterion then becomes [39]

$$\begin{aligned}
&\sum\sum(I_x(i,j;k-1)u(i,j)+I_y(i,j;k-1)v(i,j)+I(i,j;k)-I(i,j;k-1))^2\\
&\quad+\lambda\sum\sum(1-l_h(i,j))((u(i,j)-u(i,j-1))^2+(v(i,j)-v(i,j-1))^2)\\
&\quad+\lambda\sum\sum(1-l_v(i,j))((u(i,j)-u(i-1,j))^2+(v(i,j)-v(i-1,j))^2)\\
&\quad+\phi(l)
\end{aligned} \tag{144}$$

Thus a line process l is introduced, with two components l_h and l_v, so that $l_h(i,j)=1$, if an horizontal discontinuity exists between lines $j-1$ and j, and likewise for the vertical discontinuity boundaries. This line process has a cost $\phi(l)$ and can take possible *a priori* knowledge on discontinuities into account. This cost is even more justified, since the introduction of discontinuities everywhere, would lead to the indetermination of the estimation problem. In summary, the global criterion should lead to the best compromise between estimation and segmentation. Hutchinson *et al.* [39] proposed using the following cost function

$$\begin{aligned}
\phi(l)&=c_c\sum\sum l_h(i,j)+c_p\sum\sum l_h(i,j)(l_h(i,j-1)+l_h(i,j-2))\\
&\quad+c_c\sum\sum l_v(i,j)+c_p\sum\sum l_v(i,j)(l_v(i,j-1)+l_v(i,j-2))\\
&\quad+c_L\phi_L(l)+c_z\sum\sum(l_h(i,j)+l_v(i,j))(1-c(i,j))
\end{aligned} \tag{145}$$

The terms with coefficient c_c correspond to the cost of the line process itself. The terms with coefficient c_p penalize the formation of very close parallel lines. The term with coefficient c_L favours the formation of long lines with few intersections. And the last term links the velocity field discontinuities to spatial discontinuities $c(i,j)$ and penalizes the formation of discontinuities at points that do not belong to the spatial contours.

Once the criterion so defined, estimation of the velocity field and of the line process provides a solution to the estimation/segmentation problem. This time, the function to be minimized is not convex. A possible solution is the simulated annealing algorithm already evoked in Section 3.3.1, and for which experimental results are given in [46]. But its complexity makes it hardly interesting for the targeted applications. Therefore, deterministic algorithms are used to obtain the velocity field and the line process by providing the local minimum of (3.144).

Hutchinson *et al.* [39] proposed going through two steps for criterion optimization. The first step consists in minimizing energy (3.144) for a given line process. Then the criterion is quadratic and the solution is obtained through a linear equation system. Upon initialization, the line process is reset to be naught everywhere. The second step consists in determining

the line process for a given velocity field. This is achieved by local operations, by accepting all the discontinuities which reduce energy (3.144). After a few iterations, the algorithm converges towards a local minimum of the energy function.

Observation of the energy function (3.144) reveals that it can be interpreted by Gibbs distribution or a Markov random field composed of the velocity field and the discontinuity sites. This choice was explicitly made by Heitz and Bouthemy [32] [33]. They proposed an energy function different from the preceding one in the local interaction part of the Markov random field

$$\begin{aligned}
&\sum\sum(I_x(i,j;k-1)u(i,j)+I_y(i,j;k-1)v(i,j)+I(i,j;k)-I(i,j;k-1))^2\\
&\quad+\lambda\sum\sum(1-l_h(i,j))\mathrm{sgn}(\Delta_v(i,j)-\beta)(\Delta_v(i,j)-\beta)^2\\
&\quad+\lambda\sum\sum(1-l_v(i,j))\mathrm{sgn}(\Delta_h(i,j)-\beta)(\Delta_h(i,j)-\beta)^2\\
&\quad+c_z\sum\sum(l_h(i,j)+l_v(i,j))(1-c(i,j))
\end{aligned} \tag{146}$$

where Δ_h and Δ_v correspond to horizontal and vertical velocity differences, respectively.

$$\left.\begin{array}{rcl}\Delta_v &=& (u(i,j)-u(i,j-1))^2+(v(i,j)-v(i,j-1))^2\\ \Delta_h &=& (u(i,j)-u(i-1,j))^2+(v(i,j)-v(i-1,j))^2\end{array}\right\} \tag{147}$$

Heitz and Bouthemy [32] proposed using a deterministic relaxation algorithm to minimize the energy function (3.146). This algorithm is that of the "iterated conditional modes" proposed by Besag [7] and already evoked in Section 3.3.1.

An approach very different from the preceding ones was proposed by Schunck [67]. Discontinuities are detected at the level of the motion constraint equation (3.9). This equation defines a straight line in plane (u, v) (Figure 13). For a given point, we consider the inter-

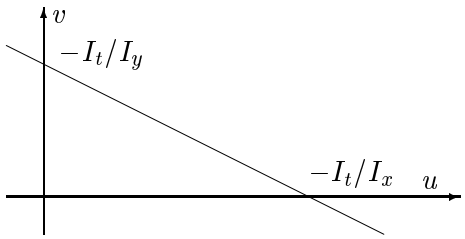

Figure 13: The motion constraint line

sections of the motion constraint line at this point, and the motion constraint lines of the neighbouring points. Surroundings within a square centred at the given point 5×5, 9×9 or 13×13 in size, are reported. Thus with a 5×5 surroundings, there will be 24 intersection points all located on the same straight line of plane (u, v). Velocity vector estimation also uses velocity field coherence, by choosing in the vicinity, the most coherent group of points, instead of all the points. This grouping is achieved by determining, on the motion constraint line of the point, the shortest interval containing half of the intersection points. The middle of this interval provides the estimate of the velocity vector of the point. On the velocity fields so estimated by using selective coherence, discontinuity detection is performed using a gradient algorithm on the two velocity field components. The last step is an iterative smoothing algorithm which takes the previously detected boundaries into account.

3.3.3 Parametric estimation

In this paragraph, we present methods for the segmentation of the velocity field when estimation is made with a parametric model. The model may be very simple, such as the two-parameter model of 2-D translation, or more sophisticated, such as the six-parameter linear model (3.132), or still a 8- or 12-parameter quadratic model (3.15).

Two categories of approaches to segmentation are to be distinguished. The first one consists in using a given structure for the region shape, either rectangular or square, relying on a tree-like structure for picture description, either binary or quaternary, and using splitting and/or merging type methods for segmentation. The second approach imposes no shape for spatio-temporal regions. These approaches are presented hereafter in that order.

Formation of the quadtree for picture description is achieved by dividing, at each level, each part of the picture into four parts of equal sizes. With a binary tree, the division is carried out successively along the horizontal and vertical axes. To apply a merging and/or splitting type algorithm, it is necessary to perform an homogeneity test on a region. Thus a region, if not satisfying the homogeneity test, is divided, and two contiguous regions are merged, if their union satisfies the test. In order to provide such a statistical test, what follows is set in a probabilistic context.

Taking hypothesis H_0 as that of spatio-temporal homogeneity of a region R, and H_1 the hypothesis of heterogeneity. We suppose that under the homogeneity hypothesis, we have

$$H_0 : I(i,j;k) = I(i,j;k-1|\theta) + n(i,j) \tag{148}$$

and under the heterogeneity hypothesis we have

$$H_1 : I(i,j;k) = I(i,j;k-1|\theta) + \mu + n(i,j) \tag{149}$$

where θ is the vector that includes all the parameters of the model to be estimated, μ is a deviation from the motion constraint equation, resulting from the heterogeneity of a region, and $n(i,j)$ is a assumed to be Gaussian noise, zero-mean and white, whose variance is unknown. To determine the test, we use the likelihood ratio method, and the ratio takes the following form

$$\frac{\max\limits_{\theta,\sigma^2} L(I(i,j;k) : (i,j) \in R|\theta,\sigma^2)}{\max\limits_{\theta,\mu,\sigma^2} L(I(i,j;k) : (i,j) \in R|\theta,\mu,\sigma^2)} \begin{array}{c} H_0 \\ > \\ < \\ H_1 \end{array} \gamma_0 \tag{150}$$

where it is implicitly supposed that the picture at moment $k-1$ is given. Under all assumptions, we obtain the following test

$$\Lambda = \frac{\sum_{(i,j)\in R}(I(i,j;k) - I(i,j;k-1|\hat{\theta}_0))^2}{\sum_{(i,j)\in R}(I(i,j;k) - I(i,j;k-1|\hat{\theta}_1) - \hat{\mu})^2} \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} \gamma \tag{151}$$

where $\hat{\theta}_0$ (resp. $\hat{\theta}_1$) are maximum likelihood estimators of θ under hypothesis H_0 (resp. H_1) and $\hat{\mu}$ is the estimator of μ. Thus, under hypothesis H_0, these estimators are as they were presented in 3.3.2. As to hypothesis H_1, this is a joint estimation of $\hat{\mu}$ and $\hat{\theta}_1$, which can be carried out iteratively as follows

$$\left.\begin{array}{rcl} \hat{\theta}_1 & = & \arg\min \sum_{(i,j)\in R}(I(i,j;k) - I(i,j;k-1|\theta_1) - \hat{\mu})^2 \\ \hat{\mu} & = & \frac{1}{\mathrm{card}[R]} \sum_{(i,j)\in R}(I(i,j;k) - I(i,j;k-1|\hat{\theta}_1)) \end{array}\right\} \tag{152}$$

Initialization is performed with $\hat{\mu} = 0$, then the two equations in (3.151) are applied iteratively until the solution converges. Minimization in the first equation in (3.151) is similar to that in hypothesis H_0. The threshold γ in (3.150) can be determined by allowing for a probability of error, such as false detection of heterogeneity, *i.e.*, an error entailing the division of a region that is in fact uniform, $Pr\{\Lambda > \gamma | H_0\}$. A common value for this error probability is 0.05. It is only necessary to know the probability distribution of Λ to determine γ. Under the previously mentioned assumptions on inter-frame difference, the variable $\frac{N-1}{N}\Lambda$ obeys to the Snedecor law with $(N, N-1)$ degrees of freedom, where N is the number of points of the region. Thus the threshold depends on the region size, and is higher as the region is smaller.

Bouthemy and Santillana Rivero [14] proposed another statistical model, by supposing as known the noise variance, to decide, using the corresponding likelihood test, whether a region is homogeneous or not. The test obtained uses a threshold unrelated to the size of the region. Nicolas and Labit [60] proposed using a threshold test that uses only the numerator of the ratio in (3.150).

In the other category of methods, no tree-like structure is imposed. Such a method has been proposed by Murray and Buxton [55], using a Markov random field to modelize the possible interpretations of the velocity field. An interpretation involves an homogeneous region characterized by the 8 parameters of (3.41) formulas. A line process is also introduced, to take into account the interactions that exist between the points pertaining to the region boundaries. Given a segmentation into spatio-temporal regions, the model parameters are estimated for each region by a least squares method. A quadratic error function measures the fitting of the parameters in relation to the measured values of the normal component on the velocity vector contour. Once the global energy function is determined, then an algorithm of simulated annealing is used for the optimization, which however has the disadvantage of being very complex. Another weak point of the method is that the number of regions has to be pre-set, but some remedies have been proposed to escape this constraint.

In the following chapters, we present picture compression methods which make use of the results of this chapter on motion detection, estimation and segmentation.

4 Appendices

4.1 Appendix 3A: Regularization method

The regularization theory permits to make well-posed an ill-posed problem. A problem is well-posed when its solution exists, is unique and continuous [72]. The last property means that the solution depends continuously on the observation. Well conditioned problems are still more interesting because numerical stability is then ensured. The following presentation is limited to the ill-posed problems concerning a linear system

$$Ax = b \tag{153}$$

where the matrix A is, either singular, or near the singularity.

The principal idea of the regularization theory consists in providing an *a priori* knowledge on the set of admissible solutions [72]. A method for obtaining a regular solution is the introduction of a quadratic criterion, which simultaneously contains the closeness to the data and a regularizing term, like the following one

$$J = \| Ax - b \|^2 + \lambda \| Px \|^2 \tag{154}$$

The matrix P renders the existing knowlegde on the physics of the problem. The quadratic criterion is chosen for mathematical simplicity reasons, but also because this corresponds to an observation disturbed by an additive white Gaussian noise.

The matrix P results from the discretization of a linear functional of the derivatives of the function, whose the discretization gives x. In the one-dimensional case often only the first derivative is used, and in the multi-dimensional case only the first partial derivatives. Thus for the 2-D case we could have the following criterion

$$J = \| Ax - b \|^2 + \lambda(\| D_1 x \|^2 + \| D_2 x \|^2) \tag{155}$$

where D_1 and D_2 are the operators of the first partial derivatives according the two principal directions.

It is obvious that the minimization of (154) or (155) is given by the solution of a system of linear equations. Let us consider the first of these minimization problems to find

$$x = (A^T A + \lambda P^T P)^{-1} A^T b \tag{156}$$

where direct or indirect, and always in practice iterative, methods permit the numerical solution of the system.

A regularizing operator, like the first derivative, used in a quadratic criterion (154) leads to smoothing, not only the perturbations, but also the discontinuities. In the presence of discontinuities, which is in practice always the case, the optimization criterion should be modified so as to relax smoothing on discontinuity points. Such criteria and methods are proposed in [11] [70].

The tutorial review [5] discusses implications of the regularization approach to many problems arising in computational vision.

4.2 Appendix 3B: Conjugate gradient method

The conjugate gradient method is an optimization method of a multi-variable function using a gradient linked to the considered function. We only consider here the case of quadratic functions

$$J(x) = \frac{1}{2} x^T Q x - bx \tag{157}$$

where the matrix Q is symmetric and positive definite, and the vector x is n-dimensional. The minimization of $J(x)$ is equivalent to the resolution of a system of linear equations

$$Qx = b \tag{158}$$

Two vectors d_1 and d_2 are called Q-orthogonal, if $d_1^T Q d_2 = 0$. In the space R^n the set of n Q-orthogonal vectors constitutes a complete base. Thus the vector x, which is solution of the equation (158) can be written

$$x = \sum_{i=0}^{n-1} \alpha_i d_i \tag{159}$$

with $\alpha_i = \frac{d_i^T Q x}{d_i^T Q d_i} = \frac{d_i^T b}{d_i^T Q d_i}; i = 0, \ldots, n-1$.

It is then possible to prove the following theorem, called the conjugate direction theorem [49].

Let $\{d_i; i = 0, \ldots, n-1\}$ be a set of n Q-orthogonal vectors. For every initial vector $x_0 \in R^n$, the sequence $\{x_k; k = 0, \ldots, n-1\}$ given by

$$x_{k+1} = x_k + \alpha_k d_k \tag{160}$$

with $\alpha_k = -\frac{(Qx-b)^T d_k}{d_k^T Q d_k}$ converges to the solution of the system (158) after n steps.

The conjugate gradient algorithm applies the above theorem by simultaneously generating the directions d_k, and it is shown that it converges in at most n steps. The algorithm follows.

Initialization: x_0, $d_0 = b - Qx_0$
For $k = 0, \ldots, n-1$, do
Correction in the direction of d_k

$$x_{k+1} = x_k + \alpha_k d_k, \text{ with } \alpha_k = \frac{(b-Qx_k)^T d_k}{d_k^T Q d_k}$$

Determination of the new direction

$$d_{k+1} = b - Qx_{k+1} + \beta_k d_k, \text{ with } \beta_k = \frac{(Qx_{k+1}-b)^T Q d_k}{d_k^T Q d_k}$$

In practice the number of iterations might exceed n because of rounding errors. The conjugate gradient algorithm is more attractive in the case of sparse matrices, because of low computation complexity, the more expensive operation being the multiplication of a square matrix by an n-dimensional vector. A sparse matrix is got in the case of the discretization by the method of finite differences, as it is the case of estimation in paragraphe 3.2.2.

4.3 Appendix 3C: Kalman filtering

The objective of Kalman filtering is the optimal estimation, minimizing the mean squared error, of the state of a system. A dynamic model of the system state is admitted, and an observation related to the system state is provided. The formulation of the model in the case of a linear system is given below. Let us note that the knowledge of the state dynamic model supposes an order on the indexes of the state. We here consider only the one-dimensional case, where order is natural, or at least it is uniquely determined. The one-dimensional space may also be constituted by any curve on the 2-D plane.

Let $X(k)$ be the n-dimensional vector describing system state. A linear difference equation describes a stochastic linear dynamic system

$$X(k+1) = \Phi(k)X(k) + G(k)w(k); k = 1, 2, \ldots \tag{161}$$

with an initial condition on the vector $X(1)$. The sequence of vectors $\{w(k)\}$ is supposed random, zero-mean and white

$$E\{w(k)w^T(j)\} = Q(k)\delta(k, j) \tag{162}$$

where $\delta(k, j)$ is Kronecker's delta. The dimension of the vector $w(k)$ may be different from that of the state vector $X(k)$. The state cannot be directly observed. However measurements linearly linked to the state are available

$$z(k) = H(k)X(k) + v(k) \tag{163}$$

The sequence of measurement noise is random, zero-mean and white

$$E\{v(k)v^T(k)\} = R(k)\delta(k, j) \tag{164}$$

In addition, state and measurement noises are not correlated.

Given the model just described, Kalman filter gives the optimal linear estimator of the state $X(k)$, which minimizes the mean squared error, knowing the measurements $\{z(j); j = 1, \ldots, k\}$. An initial condition is given on the state $X(1)$, which is supposed random with mean $\hat{X}(1|0)$ and covariance matrix $P(1|0)$. We note $\hat{X}(k|l)$ the estimator of the state $X(k)$ knowing the observations until index l. Kalman filtering is obtained in two stages: prediction and correction/filtering. For each of these stages a covariance matrix of the corresponding estimation error should be computed. Thus the following algorithm is obtained.

Initialization: $\hat{X}(1|0)$, $P(1|0)$
For $k = 1, 2, \ldots$
Filtering/correction
$\hat{X}(k|k) = \hat{X}(k|k-1) + K(k)[z(k) - H(k)\hat{X}(k|k-1)]$
with the Kalman filter gain
$K(k) = P(k|k-1)H^T(k)[H(k)P(k|k-1)H^T(k) + R(k)]^{-1}$
Covariance matrix of the estimation error
$P(k|k) = P(k|k-1) - K(k)H(k)P(k|k-1)$
$= (I - K(k)H(k))P(k|k-1)(I - K(k)H(k))^T + K(k)R(k)K^T(k)$
Prediction
$\hat{X}(k+1|k) = \Phi(k)\hat{X}(k|k)$
Covariance matrix of the prediction error
$P(k+1|k) = \Phi(k)P(k|k)\Phi^T(k) + G(k)Q(k)G^T(k)$

The computation of filter gain supposes not only the knowledge of the state transition matrix $\Phi(k)$ and the observation matrix $H(k)$, but also the knowledge of the covariance matrices of the state and measurement noises. For real-time applications this supposes learning methods for the model parameters, otherwise a conservative approach should be adopted using a low filter gain, to avoid filter divergence. There exist some techniques for verifying filter divergence by estimating the model parameters [43] [52].

For some applications we deal with in Chapter 3 for displacement estimation, we have a finite number of observations $\{z(k); k = 1, 2, \ldots, N\}$ and we have to estimate the state vector for all indexes from 1 to N. To solve this smoothing problem the above forward Kalman

filter should be used, followed by a backward filter which corrects the forward estimation. The backward filter given between others in [63] is the following

$$\hat{X}(k|N) = \hat{X}(k|k) + C(k)[\hat{X}(k+1|N) - \Phi(k)\hat{X}(k|k)], \text{for} k = N-1, \ldots, 1$$

where $C(k) = P(k|k)\Phi^T(k)P^{-1}(k+1|k)$.

Finally the system model might be non-linear. We here consider uniquely the case where the observation vector is a vector of non-linear functions of the state

$$z(k) = h(X(k); k) + v(k)$$

By linearization we obtain

$$H(k) = \left\{ \frac{\partial h_i(X(k); k)}{\partial x_j(k)} \right\}$$

where $h_i(.)$ is a component of the vector of functions, and $x_j(k)$ is a component of the state vector $X(k)$. The filtering equation is modified as follows

$$\hat{X}(k|k) = \hat{X}(k|k-1) + K(k)[z(k) - h(\hat{X}(k|k-1); k)]$$

All the other equations of the Kalman filter remain unchanged. The filter in this way obtained is called the extended Kalman filter [43].

4.4 Appendix 3D: Adaptive filtering

The objective of the adaptive filtering is the identification of the state of a system $\{X(n); n = 1, 2, \ldots\}$ given an observation, here supposed scalar, of the output of the system

$$y(n) = h(X(n); n) + z(n) \tag{165}$$

where $z(n)$ is a white noise. The criterion used is the minimization of the mean squared error, and the algorithm resulting is called the Least Mean Square (LMS) algorithm, which is described by the following equation [80]

$$\hat{X}(n) = \hat{X}(n-1) + \mu H(n)(y(n) - h(\hat{X}(n-1); n)), n = 1, 2, \ldots \tag{166}$$
$$\hat{X}(0) = 0$$

where the vector $H(n)$ is the gradient vector of the function $h(.)$, and μ is a positive constant gain, the adaptation step size.

The above described algorithm converges to a steady state under a condition on μ, provided that the function $h(.)$ is linear and the vector $H(n)$ is stationary [81]

$$\mu < \frac{2}{E\{\|H(n)\|^2\}} \tag{167}$$

The speed of convergence is proportional to the adaptation step size μ, but fluctuations also are proportional to this parameter, convergence not being strict. A good choice for the adaptation step size is the following [50]

$$\mu = \frac{1}{E\{\|H(n)\|^2\}} \tag{168}$$

Another more fast adaptive algorithm is the Normalized LMS (NLMS), which in comparison with the LMS algorithm alters the gain, more precisely the magnitude of the gain without changing its direction. Two versions may be considered

$$\hat{X}(n) = \hat{X}(n-1) + \mu \frac{H(n)}{\|H(n)\|^2}(y(n) - h(\hat{X}(n-1); n)) \tag{169}$$

with $\|H(n)\|^2 > \epsilon$, or

$$\hat{X}(n) = \hat{X}(n-1) + \mu \frac{H(n)}{\lambda + \|H(n)\|^2}(y(n) - h(\hat{X}(n-1); n)) \tag{170}$$

The last algorithm converges to a steady state provided that $0 < \mu < 2$, for a linear function $h(.)$ and for a stationary process $H(n)$ [10].

4.5 Appendix 3E: Markov and Gibbs random fields

At first we give the definitions of discrete Markov and Gibbs random fields on a lattice, which is a set of pixels

$$L = \{(m,n) : 1 \leq m \leq M, 1 \leq n \leq N\} \tag{171}$$

A Markov random field $\{X(m,n) : (m,n) \in L\}$ is defined using a local property

$$\begin{aligned} &Pr\{X(m,n) = x(m,n) | X(k,l) = x(k,l), (k,l) \neq (m,n)\} \\ &\quad = Pr\{X(m,n) = x(m,n) | X(k,l) = x(k,l), (k,l) \in V(m,n)\} \end{aligned} \tag{172}$$

where $V(m,n)$ is a set of pixels in the neighborhood of (m,n). It is supposed that the set $V(m,n)$ is defined in an homogeneous way for all pixels. A common example of neighborhood is that of 8 nearest pixels.

A Gibbs random field provides a global description of the field

$$Pr\{X(m,n) = x(m,n) : (m,n) \in L\} = \frac{e^{-U(x)}}{\Psi} \tag{173}$$

where x is a vector of MN elements. The function of MN variables $U(x)$ is called the energy function, and normalizing constant $\Psi = \sum e^{-U(x)}$ is called the partition function. An important and interesting class of Gibbs random fields is that where the energy function is composed from local interaction models. To this aim a clique is defined in a neighborhood system. A clique is a set of pixels containing, either one pixel, or neighbor pixels. Figure 14

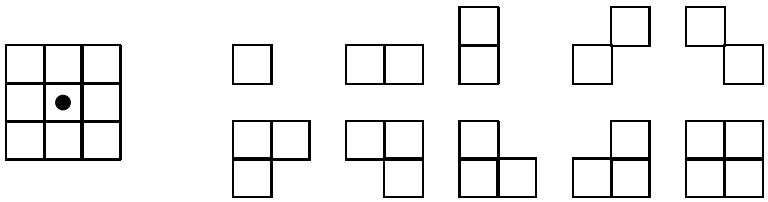

Figure 14: 8-pixel neighborhood and associated cliques

gives an 8-pixel neighborhood and the associated cliques. We are only interested on Gibbs random fields such that

$$U(x) = \sum_{c \in C} \phi_c(x) \tag{174}$$

where C is the set of all cliques on the lattice L, and $\phi_c(x)$ is called the potential of the clique c. In [6] a proof (via the Hammersley-Clifford expansion) may be found that a Gibbs random field satisfying Equation (174) is equivalent to a Markov random field. Often for the 8-pixel neighborhood, only at most pairs of pixels are considered to have a non zero potential. An example of potential for a pair of pixels, which can be used for segmentation purposes, is the following [20]

$$\phi_c(x) = \begin{cases} -\zeta_c & \text{if the two pixels have the same value} \\ \zeta_c & \text{otherwise} \end{cases} \tag{175}$$

Let us now consider the problem of estimating a Markov or Gibbs random field given an observation $\{Y(m,n) : (m,n) \in L\}$ of the field disturbed by an independent noise. It is then supposed known the following conditional probability

$$\begin{aligned} &Pr\{Y(m,n) = y(m,n) | X(k,l) = x(k,l), (k,l) \in L\} \\ &\quad = Pr\{Y(m,n) = y(m,n) | X(m,n) = x(m,n)\} \end{aligned} \tag{176}$$

The estimation problem is solved by a Bayesian approach, by obtaining the field maximizing the *a posteriori* probability

$$\begin{aligned} &Pr\{X(m,n) = x(m,n) | Y(m,n) = y(m,n); (m,n) \in L\} \\ &\quad \propto \prod_{(m,n)\in L} Pr\{Y(m,n) = y(m,n) | X(m,n) = x(m,n)\} \\ &\quad Pr\{X(m,n) = x(m,n) : (m,n) \in L\} \end{aligned} \tag{177}$$

This is also equivalent to minimize

$$\begin{aligned} &\sum_{(m,n)\in L} \phi_c(x(m,n)) \\ &\quad - \sum_{(m,n)\in L} \ln Pr\{Y(m,n) = y(m,n) | X(m,n) = x(m,n)\} \end{aligned} \tag{178}$$

The simulated annealing algorithm is used in [29] to obtain the minimum of the above function. The computational cost of this algorithm, in spite of its convergence to the global minimum of the function, prevents its use in image communication applications. Alternatives are the ICM (Iterated Conditional Modes) algorithm [7], dynamic programming [20], maximization of posterior marginals (MPM) [51], and mean field annealing [28].

A more extensive treatement of Markov and Gibbs random fields may be found in [24] and [21].

For joint estimation/segmentation problems a suitable Markov random field contains two components: one of a, possibly continuous variable, field described by a local property like that of Equation (172), or the equivalent clique potential, and a line process, for which local interactions may also be introduced [29]. The line process represents possible edge elements and the corresponding sites are placed midway between pixels on the principal directions. The line process is often considered binary, the value '1' noting the existence of an edge element. An example of cliques for the line process is shown in Figure 15 [29]; all the rotations of these cliques should also be considered.

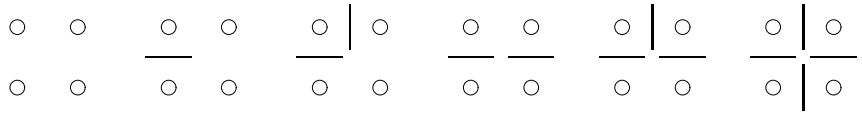

Figure 15: Cliques of the line process

4.6 Appendix 3F: Equivalence of a Kalman-based and a Wiener-based estimator

Under hypothesis (3.111) the Kalman filter gain and the covariance matrices must satisfy the following equations

- filter gain

$$K(i,j) = P^0(i,j)G(i,j)(G^T(i,j)P^0(i,j)G(i,j) + R\mathbf{I}_p)^{-1} \tag{179}$$

- covariance of estimation error

$$P(i,j) = (\mathbf{I}_2 - K(i,j)G^T(i,j))P^0(i,j) \tag{180}$$

- covariance of prediction error

$$P^0(i+1,j) = P(i,j) + \sigma_v^2 G(i,j)(G^T(i,j)G(i,j) + \mu\mathbb{I}_p)^{-1}G^T(i,j) \tag{181}$$

It is obvious that the above equations admit as solution [61]

$$P^0(i+1,j) = P^0(i,j) = \sigma_v^2\mathbb{I}_2 \tag{182}$$

From this solution results

$$K(i,j) = G(i,j)\left(G^T(i,j)G(i,j) + \frac{R}{\sigma_v^2}\mathbb{I}_p\right)^{-1} \tag{183}$$

and

$$P(i,j) = \sigma_v^2\left(\mathbf{I}_2 - G(i,j)\left(G^T(i,j)G(i,j) + \frac{R}{\sigma_v^2}\mathbf{I}_p\right)^{-1}G^T(i,j)\right) \tag{184}$$

Equation (3.112) is therefore demonstrated, and by putting $p = 1$ we obtain also (3.108) to (3.110).

References

[1] T. Aach, A. Kaup, and R. Mester, "A statistical framework for change detection in image sequences", *Colloque sur le Traitement du Signal et des Images*, pp. 1149-1152, Sept. 1991.

[2] G. Adiv, "Determining three-dimensional motion and structure from optical flow generated by several moving objects", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, pp. 384-401, July 1985.

[3] M. Askar and H. Derin, "A recursive algorithm for the Bayes solution of the smoothing problem", *IEEE Transactions on Automatic Control*, Vol. AC-26, pp. 558-561, Apr. 1981.

[4] N. Baaziz, "Approches d'estimation et de compensation de mouvement multiresolutions pour le codage de séquences d'images", Thèse, Université de Rennes, Oct. 1991.

[5] M. Bertero, T. Poggio, and V. Torre, "Ill-posed problems in early vision", *Proceedings of the IEEE*, Vol. 76, pp. 869-889, Aug. 1988.

[6] J. Besag, "Spatial interaction and the statistical analysis of lattice systems", *Journal of the Royal Statistical Society*, Vol. B-36, pp. 192-326, 1974.

[7] J. Besag, "On the statistical analysis of dirty pictures", *Journal of the Royal Statistical Society*, Vol. B-48, pp. 259-302, 1986.

[8] J. Biemond, L. Looijenkga, and D.Boekee, "A pel-recursive Wiener-based displacement estimation algorithm", *Signal Processing*, Vol. 13, pp. 399-412, Dec. 1987.

[9] M. Bierling and R. Thoma, "Motion compensating field interpolation using a hierarchically structured motion estimator", *Signal Processing*, Vol. 11, pp. 387-404, 1986.

[10] R. Bitmead and B. Anderson, "Performance of adaptive estimation algorithms in dependent random environments", *IEEE Transactions on Automatic Control*, Vol. AC-25, pp. 788-794, Aug. 1980.

[11] A. Blake and A. Zisserman, *Visual reconstruction*, MIT Press, 1987.

[12] P. Bouthemy, "Estimation et structuration d'indices spatio-temporels pour l'analyse du mouvement dans une séquence d'images", *Traitement du Signal*, Vol. TS-4, pp. 239-257, 1987.

[13] P. Bouthemy, "A maximum-likelihood framework for determining moving edges", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-11, pp. 499-511, May 1989.

[14] P. Bouthemy and J. Santillana Rivero, "A hierarchical likelihood approach for region segmentation according to motion-based criteria", *Proceedings of the Inter. Conf. on Computer Vision*, pp. 463-467, June 1987.

[15] S. Brofferio, M. Monti, V. Rampa, and M. Taliercio, "VLSI recursive motion estimator chip set", *Proceedings of the IEEE Inter. Symposium on Circuits and Systems*, pp. 57-60, 1990.

[16] B. Buxton and H. Buxton, "Computation of optic flow from motion of edge features in image sequences", *Image and Vision Computing*, Vol. 2, pp. 59-75, May 1984.

[17] C. Cafforio and F. Rocca, "Methods for measuring small displacements of television images", *IRE Transactions on Information Theory*, Vol. IT-22, pp. 573-579, Sept. 1976.

[18] C. Cafforio and F. Rocca "The differential method for motion estimation", in: T. S. Huang, ed., *Image Sequence Processing and Dynamic Scene Analysis*, pp. 104-124, Springer-Verlag, 1983.

[19] N. Cornelious and T. Kanade, "Adapting optical flow to measure object motion in reflectance and X-ray image sequences", *ACM SIGGRAPH/SIGART Interdisciplinary Workshop on Motion*, pp. 50-58, Apr. 1983.

[20] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, pp. 39-55, Jan. 1987.

[21] H. Derin and P. Kelly, "Discrete-index Markov-type random processes", *Proceedings of the IEEE*, Vol. 77, pp. 1485-1510, Oct. 1989.

[22] J. Driessen and J. Biemond, "Motion field estimation by 2-D Kalman filtering", *European Signal Processing Conference*, pp. 975-978, Sept. 1990.

[23] J. Driessen, L. Böröczky, and J. Biemond, "Pel-recursive motion field estimation from image sequences", *Journal of Visual Communication and Image Representation*, Vol. 2, pp. 259-280, Sept. 1991.

[24] R. Dubes and A. K. Jain, "Random field models in image analysis", *Journal of Applied Statistics*, Vol. 16, No. 2, pp. 131-164, 1989.

[25] J.-L. Dugelay and H. Sanson, "Analyse de mouvements globaux dans les séquences d'images et application au codage de télévision", *Colloque sur le Traitement du Signal et des Images*, pp. 881-884, Sept. 1991.

[26] G. D. Forney, "The Viterbi algorithm", *Proceedings of the IEEE*, Vol. 61, pp. 268-278, Mar. 1973.

[27] E. Frimout, J. Driessen, and E. Deprettere, "Parallel architecture for a pel-recursive motion estimation algorithm", *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 2, pp. 159-168, June 1992.

[28] D. Geiger and F. Girosi "Parallel and deterministic algorithms from MRF's: surface reconstruction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-13, pp. 401-412, 1991.

[29] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-6, pp. 721-741, Nov. 1984.

[30] B. Haskell, F. Mounts, and J. Candy, "Interframe coding of videotelephone pictures", *Proceedings of the IEEE*, Vol. 60, pp. 792-800, July 1972.

[31] D. Heeger, "Optical flow using spatio-temporal filters", *International Journal of Computer Vision*, Vol. 1, pp. 279-302, Jan. 1988.

[32] F. Heitz and P. Bouthemy, "Motion estimation and segmentation using global Bayesian approach", *Proceedings of the IEEE Inter. Conf. on Acoustics, Speech, and Signal Processing*, pp. 2305-2308, Apr. 1990.

[33] F. Heitz and P. Bouthemy, "Multimodal estimation of discontinuous optical flow using Markov random field", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-15, pp. 1217-1232, Dec. 1993.

[34] E. Hildreth, *The measurement of visual motion*, MIT Press, 1983.

[35] M. Hoetter, "Differential estimation of the global motion parameters zoom and pan", *Signal Processing*, Vol. 16, pp. 249-265, Mar. 1989.

[36] B. Horn, *Robot vision*, MIT Press, 1986.

[37] B. Horn and B. Schunck, "Determining optical flow", *Artificial Intelligence*, Vol. 17, pp. 185-203, Aug. 1981.

[38] Y. Hsu, H.-H. Nagel, and G. Rekkers, "New likelihood test methods for change detection in image sequences", *Computer Vision, Graphics, and Image Processing*, Vol. 26, pp. 73-106, 1984.

[39] J. Hutchinson, C. Koch, T. Luo, and C. Mead, "Computing motion using analog and binary resistive networks", *IEEE Computer*, pp. 52-63, Mar. 1988.

[40] L. Jacobson and H. Wechsler, "Derivation of optical flow using a spatiotemporal frequency approach", *Computer Vision, Graphics, and Image Processing*, Vol. 38, pp. 29-65, 1987.

[41] J. R. Jain and A. K. Jain, "Displacement measurement and its application in interframe image coding", *IEEE Transactions on Communications*, Vol. COM-29, pp. 1799-1808, Dec. 1981.

[42] R. Jain, D. Militzer, and H.-H. Nagel, "Separating non-stationary from stationary scene components in a sequence of real world TV-images", *Proc. of the IJCAI*, pp. 612-618, 1977.

[43] A. Jazwinski, *Stochastic processes and filtering theory*, Academic Press, 1970.

[44] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated interframe coding for video conferencing", *Proceedings of the NTC*, pp. G5.3.1-G5.3.5, 1981.

[45] T. Komarek and P. Pirsch, "Array architectures for block matching algorithms", *IEEE Transactions on Circuits and Systems*, Vol. CAS-36, pp. 1301-1308, Oct. 1989.

[46] J. Konrad and E. Dubois, "Bayesian estimation of motion vector fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-14, pp. 910-927, Sept. 1992.

[47] C. Labit and A. Benveniste, "Motion estimation in a sequence of television pictures", in: T. S. Huang, ed., *Image Sequence Processing and Dynamic Scene Analysis*, pp. 292-306, Springer-Verlag, 1983.

[48] P. Lalande, "Détection du mouvement dans une séquence d'images selon une approche markovienne; application à la robotique sous-marine", Thèse, Université de Rennes, Mar. 1990.

[49] D. Luenberger, *Introduction to linear and nonlinear programming*, Addison-Wesley, 1973.

[50] O. Macchi and M. Bellanger, "Le filtrage adaptatif transverse", *Traitement du Signal*, Vol. TS-5, pp. 115-132, 1988.

[51] J. Marroquin, S. Miter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision", *Journal of the American Statistical Association*, Vol. 82, pp. 76-89, 1987.

[52] R. Mehra, "On the identification of variances and adaptive Kalman filtering", *IEEE Transactions on Automatic Control*, Vol. AC-15, pp. 175-184, Apr. 1970.

[53] M. Mijiyawa, "Utilisation de l'information de mouvement pour amélioration de la définition des images de télévision couleur", Thèse Docteur-Ingénieur, Ecole Nat. Sup. de Télécommunications, Paris, 1987.

[54] R. Moorhead, S. Rajala, and L. Cook, "Image sequence compression using a pel-recursive motion-compensated method", *IEEE Journal on Selected Areas in Communications*, Vol. SAC-5, pp. 1100-1114, Aug. 1987.

[55] D. Murray and B. Buxton, "Scene segmentation from visual motion using global optimization", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, pp. 220-228, Mar. 1987.

[56] H.-G. Musmann, P. Pirsch, and H.-J. Grallert, "Advances in picture coding", *Proceedings of the IEEE*, Vol. 73, pp. 523-548, Apr. 1985.

[57] H.-H. Nagel, "On the estimation of optical flow: relation between different approaches and some new results", *Artificial Intelligence*, Vol. 33, pp. 299-324, 1987.

[58] A. Netravali and J. Robbins, "Motion-compensated television coding: Part I", *Bell System Technical Journal*, Vol. 58, pp. 631-670, 1979.

[59] A. Netravali and J. Salz, "Algorithms for estimation of three-dimensional motion", *ATT Journal*, Vol. 64, pp. 335-345, Feb. 1985.

[60] H. Nicolas and C. Labit, "Global motion identification for image sequence analysis and coding", *Picture Coding Symposium*, Sept. 1991.

[61] J.-C. Pesquet, "Méthodes adaptatives en codage prédictif des images numériques", Thèse, Université Paris-Sud, 1990.

[62] I. Petrovski, *Ordinary differential equations*, Prentice-Hall, 1966.

[63] H. Rauch, F. Tung, and C. Striebel, "Maximum likelihood estimates of linear dynamic systems", *Amer. Inst. of Aeronautics Journal*, Vol. 3, pp. 1445-1450, Aug. 1965.

[64] C. Reillo, "Contour-based displacement estimation with application to picture coding", Thesis, School of Electrical Engineering, Stockholm, 1989.

[65] A. Rougée, B. Levy, and A. Willsky, "Reconstruction of two-dimensional velocity fields as a linear estimation problem", *Proceedings of the Inter. Conf. on Computer Vision*, pp. 646-650, 1988.

[66] J. Santillana Rivero, "Etude du codage MIC différentiel adaptatif de séquences d'image de télévision avec compensation du mouvement", Thèse Docteur-Ingénieur, INSA de Rennes, 1985.

[67] B. Schunck, "Image flow segmentation and estimation by constraint line clustering", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-11, pp. 1010-1027, Oct. 1989.

[68] R. Srinivasan and K. Rao, "Predictive coding based on motion estimation", *IEEE Transactions on Communications*, Vol. COM-33, pp. 888-896, Aug. 1985.

[69] J. Stuller and G. Krishnamurthy, "Kalman filter formulation of low-level television image motion estimation", *Computer Vision, Graphics, and Image Processing*, Vol. 21, pp. 169-204, 1983.

[70] D. Terzopoulos, "Regularization of inverse visual problems involving discontinuities", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, pp. 413-424, July 1986.

[71] W. Thompson, K. Mutch, and V. Berzins, "Dynamic occlusion analysis in optical flow fields", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-7, pp. 374-383, Jul. 1985.

[72] A. Tikhonov and V.Arsenin, *Solutions of ill-posed problems*, Winston and Sons, 1977.

[73] G. Tziritas, "Recursive and/or iterative estimation of two-dimensional velocity field and reconstruction of three-dimensional motion", *Signal Processing*, Vol. 16, pp. 53-72, Jan. 1989.

[74] G. Tziritas, "Smoothing the displacement field for edge-based motion estimation", *Proceedings of the European Signal Processing Conference*, pp. 697-670, Sept. 1990.

[75] S. Ullman, "Analysis of visual motion by biological and computer systems", *IEEE Computer*, Vol. 14, pp. 57-69, Aug. 1981.

[76] A. Verri and T. Poggio, "Motion field and optical flow: qualitative properties", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-11, pp. 490-498, May 1989.

[77] L. Vos and M. Stegherr, "Parametrizable VLSI architecures for the full-search block-matching algorithm", *IEEE Transactions on Circuits and Systems*, Vol. CAS-36, pp. 1309-1316, Oct. 1989.

[78] D. Walker and K. Rao, "Improved pel-recursive motion compensation", *IEEE Transactions on Communications*, Vol. COM-32, pp. 1128-1134, Oct. 1984.

[79] A. Waxman and K. Wohn, "Contour evolution, neighborhood deformation and global image flow: planar surfaces in motion", *Robotics Research*, Vol. 4, pp. 95-108, 1985.

[80] B. Widrow and M. Hoff, "Adaptive switching circuits", *IRE WESCON Conv. Rec.*, pp. 96-104, 1960.

[81] B. Widrow, J. McCool, M. Larimore, and R. Johnson, "Stationary and nonstationary learning characteristics of the LMS adaptive filter" *Proceedings of the IEEE*, Vol. 64, pp. 1151-1162, Aug. 1976.

[82] K. Wohn, J. Wu, and R. Brockett, "A contour-based recovery of image flow: iterative transformation method", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-13, pp. 746-760, Aug. 1991.

[83] K. Yang, M. Sun, and L. Wu, "A family of VLSI designs for the motion compensation block-matching algorithm", *IEEE Transactions on Circuits and Systems*, Vol. CAS-36, pp. 1317-1325, Oct. 1989.