Internationnal Journal of Visual
Communication and Image Representation
Vol. 6, No. 4, December, pp 348-365, 1995

Robust multiresolution estimation of parametric motion models

J.M. Odobez and P. Bouthemy

(Jean-Marc.Odobez@univ-lemans.fr, Patrick.Bouthemy@irisa.fr)

Abstract

This paper describes a method to estimate parametric motion models. Motivations for the use of such models are on one hand their efficiency, which has been demonstrated in numerous contexts such as estimation, segmentation, tracking and interpretation of motion, and on the other hand, their low computational cost compared to optical flow estimation. However, it is important to have the best accuracy for the estimated parameters, and to take into account the problem of multiple motion. We have therefore developed two robust estimators in a multiresolution framework. Numerical results support this approach, as validated by the use of these algorithms on complex sequences.

1 Introduction

One of the major areas in computer vision research is dynamic scene analysis, which has motivation from numerous applications ([1, 28, 27]). Some of them (meteorology, bio-medical, ...) are concerned with natural physical phenomena, and therefore, deal with scenes including non-rigid objects with fuzzy dynamical behaviour. We

can find complex situations in other domains, such as autonomous system navigation in unknown environments or telesurveillance, where human body motion is present. Achieving motion analysis in such contexts is a difficult task which requires a suitable and efficient formulation. In this paper, we perform the motion analysis by identifying 2D parametric models of the optical flow field ; in particular we use polynomial models of the point coordinates (x, y) in the image plane. Those models include constant flow (global translation), affine flow (first order polynomials in x and y), quadratic flow. This choice turned out to be judicious in many different situations such as segmentation of the image into regions with homogeneous apparent motion [35, 9], extraction and coding of temporal information in a motion-compensated coding scheme [16, 30], apparent motion estimation [36], tracking [26], and recovery of useful 3D qualitative [9] or quantitative [29] motion information. Similar models have been used with success for the matching of stereo images [2]. There are two arguments for the use of such an approach. The first one is that a small number of parameters (six in the case of affine flow) are enough to completely describe the flow vector at any point in the region of validity, which can be large, and that those flow vectors constitute a very good approximation of the real optical flow, as the studies mentioned above show. The second argument is the low computation cost.

However, obtaining reliable and accurate estimations is crucial, for instance to separate more easily different motions, or to make use of the numerical results in a second stage (for example [29]) whose efficiency usually deeply depends on the accuracy of these results. In numerous dynamic scene analysis issues, it is useful and often even necessary to first recover the motion due to camera movement, and then, to perform detection and tracking of moving objects in the scene. We present in this paper two robust multiresolution algorithms for the estimation of parametric motion models.

It is now well-known that the use of multiresolution schemes improve considerably motion analysis estimation using differential methods, i.e. using spatio-temporal gradients of intensity. This has been mainly studied and validated in dense optical flow field estimation [3, 13, 15, 23]. Accurate estimations can be recovered even with large displacements or with an irregular intensity gradient distribution in the image. More recently, [4] and [26] propose a multiresolution "least mean squares" motion parameters estimation technique. However, this technique is efficient if the region which forms the estimation support is sufficiently large. Thus, if no segmentation is available, the *a priori* region where the estimation is performed (e.g. the whole image or blocks of reasonable size) may contain several motions and the results will surely be affected. In [4], it is postulated that the global motion results from egomotion, and that the projections of moving objects occupy only a very small part of the image. There is less constraint in [5], where transparent motion is under consideration, but only constant models are used. In [26], the regions involved in the estimation process are given by a motion segmentation that ensures

a single motion per region.
The problem of the support region is therefore of great importance, given that we want to deal with complex situations or to get rid of a preliminary segmentation step, which is usually a computationally heavy and difficult task. The class of robust estimators [19, 32], which has become popular in image processing [25, 22] offers an appealing direction of investigation. Recent work has already dealt with the robust estimation of dense optical flow field [6], or of 3D structure and motion of objects in the scene, [24, 34]. Also, robust estimation approach has been used in [12, 7] for a model-based motion estimation. In [12], only a 3D translation with constant depth, i.e. a second order 2D motion model with three parameters, at a single resolution, has been considered. The method proposed in [7] for the robust estimation of affine motion models, which has been developped in parallel to our work, utilizes a different minimisation technique. A robust estimator well-adapted to the given problem must take into account three important features : data are noisy, the models we use are only approximations, and the computation cost should be as low as possible.

This paper is organized as follows. In Section 2, we describe our motion model and the objective function used in the minimization. Section 3 presents the multiresolution least-mean-squares estimation algorithm, to which our original solutions will be compared. Section 4 makes a brief review of robust estimators, and describes the method we have designed. We will show results obtained on sequences involving simulated and real motion in Section 5. Finally, Section 6 contains concluding remarks.

2 Motion model and objective function

We consider the class of 2D polynomial motion models. Using matrix notation, these models can always be stated in the following general way:

$$V_A(X_i) = \begin{bmatrix} u(X_i) \\ v(X_i) \end{bmatrix} = B(X_i)A, \tag{1}$$

which is linear with respect to the n motion parameters $A^t = (a_1, .., a_j, .., a_n)$, and where $X_i = (x_i, y_i)$ denotes the spatial image position of a point, $V_A(X_i)$ the flow vector modeled at point X_i ; B is a matrix, whose form depends on the chosen model, but whose coefficients depend only on the point coordinates.

Every model of this class can be used - constant, affine, or quadratic, complete or not - but here we will mainly deal with the complete affine model defined as :

$$\begin{cases} u(X_i) = a_1 + a_2\, x_i + a_3\, y_i \\ v(X_i) = a_4 + a_5\, x_i + a_6\, y_i \end{cases} \tag{2}$$

In this case we have :

$$B(X_i) = \begin{bmatrix} 1 & x_i & y_i & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & x_i & y_i \end{bmatrix}$$

This model is in fact a good tradeoff between complexity and representativeness. It can take into account many kinds of motion (translation, rotation, scaling, deformation), and even if a rigid 3D motion gives rise to a quadratic model (at least) in the image plane, the affine flow recovers the essential part [9, 29].
For each point X_i, we can write using vector notation:

$$\frac{dI}{dt}(X_i, t) = S(X_i, t) = \vec{V}(X_i, t).\vec{\nabla} I(X_i, t) + I_t(X_i, t)$$

where $\vec{V}^t = (\frac{dx}{dt}, \frac{dy}{dt})$ denotes the flow vector function, $\vec{\nabla} I^t = (I_x, I_y)$ and I_t are respectively the spatial gradient and the temporal derivative of the intensity function I. $S(X_i, t)$ represents the total derivative of I with respect to time t, that is to say, the instantaneous temporal variation of the intensity of the moving projected point along its planar trajectory. The constant brightness assumption, i.e. $S(X_i, t) = 0$ [18], leads to the well-known motion constraint equation :

$$\vec{V}(X_i, t).\vec{\nabla} I(X_i, t) + I_t(X_i, t) = 0.$$

However, global illumination changes can occur, for instance in outdoor scenes, or in satellite sequences (where, for instance, the frame rate is two images per hour for Meteosat) in both visible and infrared channels (in the latter case, changes are due to the diurnal and interdiurnal variability of the brightness temperatures, [33]). Hence, to deal with those changes, we choose S constant over the given region, that is:

$$S(X_i, t) = \frac{dI}{dt}(X_i, t) = -\xi \ . \tag{3}$$

Thus, we have one more parameter to estimate. Letting r_i be the following expression (dropping the time variable t when no confusion is possible):

$$\begin{aligned} r_i &= \vec{V}(X_i).\vec{\nabla} I(X_i) + I_t(X_i) - S(X_i) \\ &= I_x(X_i)\, u(X_i) + I_y(X_i)\, v(X_i) + I_t(X_i) + \xi \end{aligned} \tag{4}$$

and considering now not any general function $\vec{V}$ but the model function $\vec{V_A}$, we finally get for r_i (using matrix notation):

$$r_i = \mathcal{X}_i\, \Theta - \mathcal{Y}_i \tag{5}$$

$$\text{where } \begin{cases} \Theta^t = (A^t, \xi) \\ \mathcal{Y}_i = -I_t(X_i) \quad \text{and} \\ \mathcal{X}_i \equiv \mathcal{X}(X_i,\, I_x(X_i),\, I_y(X_i)) = (\ \nabla I(X_i)^t\, B(X_i)\ ,\ 1\ ) \end{cases}$$

In order to estimate the parameter Θ, we can minimize the following quadratic error measure:

$$E_1(\Theta) = \sum_{X_i \in W} r_i^2 = \sum_{X_i \in W} (\mathcal{X}_i\, \Theta - \mathcal{Y}_i)^2 \tag{6}$$

with respect to Θ, where W denotes the region (called the estimation support region) over which minimization is performed (e.g., the whole image or a block). We obtain the well-known solution:

$$\widehat{\Theta} = \left[\sum_{X_i \in W} \mathcal{X}_i{}^t \, \mathcal{X}_i \right]^{-1} \sum_{X_i \in W} \mathcal{X}_i{}^t \, \mathcal{Y}_i \tag{7}$$

3 Multiresolution least mean squares estimation

When the displacements[1] between two frames are too important with respect to the spatial frequency content of the image, the motion constraint equation (4) is no longer valid. The use of a multiresolution algorithm allows us to deal with large motion. In this section, we briefly recall the multiresolution least mean square estimation scheme described in [4, 26], to which we have added the estimation of the intensity parameter ξ introduced in the previous section. Let us now consider the following expression corresponding to the model-based displaced frame difference:

$$\mathrm{DFD}_\Theta(X_i) = I(X_i + B(X_i)A,\ t+1) - I(X_i, t) + \xi \tag{8}$$

Considering again the assumption introduced in (3) and approximating the total derivative of the intensity function by a finite difference, we can state the problem of estimating the parameter Θ as the minimisation of the following cost function:

$$E(\Theta) = \sum_{X_i \in W} (\mathrm{DFD}_\Theta(X_i))^2 \tag{9}$$

The use of an incremental scheme substitutes the minimisation of a series of succesive approximations of the function $E(\Theta)$, easier to be handled, for the direct minimisation of the non-linear function $E(\Theta)$.

3.1 Incremental estimation

The following development concerns the incremental estimation step at one resolution level as well as from one level to the next. The principle of this incremental scheme is presented in Figure 4 in the monodimensional case. Let $\widehat{\Theta}_k^t = (\widehat{A}_k^t, \widehat{\xi}_k)$ be the current estimate of Θ. In this case, we can write:

$$\Theta = \widehat{\Theta}_k + \Delta\Theta_k, \text{ or more precisely } \left\{ \begin{array}{c} A = \widehat{A}_k + \Delta A_k \\ \xi = \widehat{\xi}_k + \Delta\xi_k \end{array} \right.$$

and then:

$$\delta X_i = B(X_i)A = B_i\widehat{A}_k + B_i\Delta A_k \ \ (\text{ with } B_i = B(X_i)).$$

[1] Let us node that the displacement of the point X_i between two successive frames is $\delta X_i = V_A(X_i)\delta t$, where δt is the time interval between two frames. We will take $\delta t = 1$ to simplify notation.

A first order expansion of I around point $X_i + B_i\widehat{A}_k$ at time $t+1$ is performed in the expression $\mathrm{DFD}_\Theta(X_i)$, leading to the error variable r'_i given by:

$$\begin{aligned} r'_i &= I(X_i + B_i\widehat{A}_k, t+1) - I(X_i, t) + \widehat{\xi}_k \\ &\quad + \nabla I^t(X_i + B_i\widehat{A}_k, t+1)\, B_i \Delta A_k + \Delta\xi_k \\ &= \mathcal{X}'_i\, \Delta\Theta_k - \mathcal{Y}'_i \end{aligned} \tag{10}$$

$$\text{where } \begin{cases} \mathcal{Y}'_i = I(X_i, t) - I(X_i + B_i\widehat{A}_k, t+1) - \widehat{\xi}_k \\ \mathcal{X}'_i = \mathcal{X}(X_i,\, I_x(X_i + B_i\widehat{A}_k, t+1),\, I_y(X_i + B_i\widehat{A}_k, t+1)) \end{cases} \tag{11}$$

We can obtain an estimation of $\Delta\Theta_k$ by minimizing the following error function:

$$E_2(\Delta\Theta_k) = \sum_{X_i \in W} (r_i{}')^2 \tag{12}$$

which represents an approximation of the error function $E(\Theta)$ around $\Theta = \hat{\Theta}_k$. Since E_2 is linear with respect to $\Delta\Theta_k$, it can be minimized easily. In relation (11), values of I, I_x and I_y for points which are not on the image grid are computed using bilinear interpolation.

3.2 Coarse-to-fine estimation

The incremental estimation is embedded in a coarse-to-fine refinement scheme. We build a L-level low-pass Gaussian pyramid of each image as described in [10]. As a result, a displacement (measured in pixels) at level $l+1$ is half the corresponding displacement at level l. The estimation process is the following. At the coarsest level $L-1$, no prior estimation is available and we therefore minimize the error measure $E_1(\Theta)$ (relation (6). At this level, displacements are small, and hence the motion constraint equation is usually valid. Hence, a first estimate of Θ^{L-1} is obtained and successive refinements by minimizing (12) are performed at the same level until the incremental estimate $\widehat{\Delta\Theta}^{L-1}$ is too small or a given number of iterations is reached. Then, the estimated parameter $\widehat{\Theta}^{L-1}$ is projected to the finer level, where the refinement process starts again. This is repeated until the finest level is processed. The final estimate of Θ, $\widehat{\Theta_{est}}$, is therefore the value $\widehat{\Theta}^0$ obtained after the last iteration at level 0. This corresponds to the multiresolution least mean square algorithm (MRLS) summarized in Figure 1.

Different expressions of the "norm" $\|\widehat{\Delta\Theta}\|$ used in the stopping test could be considered. To save computation, we use a linear combination of the motion parameters $\widehat{\Delta a}_j$:

$$\kappa = \sum_{j=1}^{n} s_j\, |\widehat{\Delta a}_j| \tag{13}$$

where n is the number of motion parameters. Coefficients s_j are defined as follows. If $\vec{V}_{\widehat{\Delta a}_j}$ denotes the flow field supplied only by $\widehat{\Delta a}_j$ setting the other parameters to zero, we can write :

$$\frac{1}{S_W} \sum_{X_i \in W} \|\vec{V}_{\widehat{\Delta a}_j}(X_i)\| = s_j\, |\widehat{\Delta a}_j| \tag{14}$$

$\widehat{\Theta^{L-1}} \longleftarrow$ Least mean squares (residual(5))
FOR l = level $L-1$ to level 0 DO
    iter $\longleftarrow 0$
    DO
        $\widehat{\Delta\Theta^l} \longleftarrow$ Least mean squares (E_2)
        $\widehat{\Theta^l} \longleftarrow \widehat{\Theta^l} + \widehat{\Delta\Theta^l}$ and iter $\longleftarrow$ iter $+1$
    WHILE (iter $< \lambda$ and $\|\widehat{\Delta\Theta^l}\| > \frac{d}{2^l}$)
    IF $l \neq 0$: $\widehat{A^{l-1}} \longleftarrow P(\widehat{A^l})$, $\widehat{\xi^{l-1}} \longleftarrow \widehat{\xi^l}$
END FOR
$\widehat{\Theta_{est}} \longleftarrow \widehat{\Theta^0}$

Figure 1: Multiresolution least mean square algorithm (MRLS)

where S_W is the size of the estimation support region W. Considering for instance an affine motion model, we obtain :

$$s_1 = s_4 = 1,\ s_2 = s_5 = \frac{1}{S_W} \sum_{X_i \in W} |x_i - x_W|,\ s_3 = s_6 = \frac{1}{S_W} \sum_{X_i \in W} |y_i - y_W| \qquad (15)$$

where (x_W, y_W) is the gravity center of the region W. Thus, coefficient s_j can be directly related to the size of the support region, which depends on the resolution level. Therefore, the stopping criterion used in the MRLS algorithm (i.e. $\|\widehat{\Delta\Theta^l}\| > \frac{d}{2^l}$) tests if the estimated values of the incremental parameters bring a significant modification of the model flow field. The quantity d can be assimilated to a displacement (we take $d = 0.1$), and the denominator 2^l makes the test homogeneous w.r.t. the scale.

Of course, elements in (12) (i.e. I, X_i, W,...) are considered at current level l. The projection operator P introduced in Fig. 1 performs the transformation of the motion parameters from a given level to the next finer level, that is:

$$P\ : a^{l-1}_{const} \longleftarrow 2a^l_{const}\ ,\ a^{l-1}_{lin} \longleftarrow a^l_{lin}\ ,\ a^{l-1}_{quad} \longleftarrow \frac{1}{2} a^l_{quad} \qquad (16)$$

4 Robust multiresolution estimation

In statistical analysis, the goal of robust estimation is to find the parameter vector $\widehat{\Theta}$ which best fits a model $M(X_i, \Theta)$ to the observations y_i, when data X_i deviate from the statistical error distribution, i.e., when some data behave like outliers.

4.1 Review of robust estimators

Three measures are usually used to characterize robust estimators: efficiency, i.e., the ability to reach optimal estimates given a certain noise distribution, the breakdown

point, roughly defined as the highest percentage of outliers that an estimator can tolerate, and computational complexity. We now briefly present the two mainly used classes of robust estimators, [19, 32].

- M-estimator: It is defined as follows:

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \sum_i \rho(y_i - M(\Theta, X_i), \sigma) \tag{17}$$

where σ is a scale factor. The function ρ is called an M-estimator since this minimization corresponds to the Maximum-likelihood estimation, if ρ is interpreted as the opposite of the conditional log-likelihood of the observations.
The influence function, introduced in ([14]) is a tool to analyze the robustness of M-estimators. The influence function characterizes the bias that a particular error is likely to induce on the solution. In the continuous case, it corresponds to the derivative, ψ, of the ρ function. For example, we have for the least mean squares estimator : $\rho(x) = x^2$, and then $\psi(x) = 2x$. In this case, the influence of outliers increases linearily without bounds. If we replace the quadratic norm by the absolute norm, $\rho(x) = |x|$, then $\psi(x) = sign(x)$. The influence of gross errors is reduced, but the asymptotic breakdown point remains 0, which means that one single datum is likely to completely disturb the estimation. Since we want to eliminate the contribution of outliers, a hard redescending norm is appropriate. This is the case for Tukey's biweight estimator [17]. This function is plotted along with its associated influence function in Figure 5, where the ψ function is given by :

$$\psi(x, C) = \begin{cases} x(C^2 - x^2)^2 & \text{if } |x| < C\,, \\ 0 & \text{otherwise} \end{cases} \tag{18}$$

Hard redescending estimators can have a breakdown point strictly greater than 0, but it can be equal at most to $\frac{1}{p+1}$, where p is the total number of parameters of the model.

- The least-median-of-squares estimator (LMedS) : The parameters are estimated by solving the nonlinear minimization problem :

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}}\ \operatorname{Med}_i\,(\,y_i - M(\Theta, X_i)\,)^2 \tag{19}$$

Its main advantage lies in its theoretical high robustness, since it remains reliable up to 50% of the data as outliers. However, it has several drawbacks, more precisely :

  - the computation cost is very high, increasing rapidly with the amount of data, even if a Monte-Carlo-like speed-up technique is used, [25] ;
  - its efficiency, in the case of Gaussian noise, is very low, since at each iteration, a number of observations equal to the number of parameters is

employed to compute a possible estimate. Nevertheless, this efficiency can be improved by performing a least-square estimation after the removal of outlier data [25]. Besides, this way of computing parameters is also quite questionable when dealing with inexact models. The case of non-linear model is also not straightforward to handle, but solutions have been proposed, e.g. in [24] for the estimation of 3D camera location and attitude.

In our case, data in the support region W are numerous, data noise may be important (due to image acquisition noise, interpolation step, computation of intensity derivatives, ...). In addition, the model, whichever we choose (linear or quadratic), is still only an approximation of the real motion. Those considerations, and early experiments performed with real images and motions using the LMedS estimator, have led us to prefer the M-estimator method with Tukey's biweight function.

4.2 Proposed robust multiresolution estimation methods

Iteratively Reweighted Least Squares (IRLS) is a well known method to solve the M-estimation problem [17]. It converts this M-estimation problem into an equivalent weighted least-squares problem :

$$\sum_i \rho(\, r_i\,) = \sum_i \frac{1}{2}\, w_i\, r_i^2 \quad \text{with} \quad r_i = y_i - M(\Theta, X_i) \tag{20}$$

A necessary condition for minimization is that the derivates of the error measure with respect to each component Θ_j of the parameter vector Θ are null. We get :

$$\sum_i \psi(r_i)\, \frac{\partial r_i}{\partial \Theta_j} = \sum_i w_i\, r_i\, \frac{\partial r_i}{\partial \Theta_j} = 0, \quad j = 1, \ldots, p \tag{21}$$

where p is the number of parameters. Thus, the weights w_i at each point X_i are given by:

$$w_i = \frac{\psi(r_i)}{r_i}. \tag{22}$$

For a given initial value of Θ, IRLS first consists in evaluating the residuals r_i, and consequently, the weights w_i. Then, a new estimate of Θ is computed using the weighted least-squares technique. The weights are updated, and the process is repeated until convergence. When no initial value is provided, the initial weights are set to 1. In our case, this method will be embedded in a multiresolution scheme that we now describe.

The error measure E we have considered in Section 3 (see (9)) is simply reformulated here as :

$$E_r(\Theta) = \sum_{X_i \in W} \rho(\, \mathrm{DFD}_\Theta(X_i),\ C) \tag{23}$$

where ρ is the Tukey's biweight function and $\mathrm{DFD}_\Theta(X_i)$ is given in (8). The minimisation of this error is performed using the same incremental and multiresolution

scheme than in the MRLS algorithm described in section 3.2. Thus, at each step, the increment value $\Delta\Theta_k$ is estimated by minimizing the following error function:

$$E_3(\Delta\Theta_k) = \sum_{X_i \in F} \rho(r'_i,\ C) \tag{24}$$

where r'_i is given by (10). Since this residual is linear w.r.t. $\Delta\Theta_k$, the IRLS procedure can be applied straightforwardly. As a matter of fact, $\hat{\Theta}_k$ can be supposed to be not too far from the optimal solution; thus, 0 is taken as initial value for $\Delta\Theta_k$, and is used to compute the initial weights in the IRLS procedure.

However, the introduction of the robust estimator increases the number of local minima of the error function. To alleviate this problem, we have adopted a scheme similar to GNC (Graduated Non Convexity) [8]. When computing the first estimation of Θ at the coarsest level (where in fact residual of equation (5) is used), all the weights are set to 1 in the IRLS. Then, outliers are gradually eliminated by controlling the value of C, which characterizes the shape of the robust function. At the beginning of the process, the estimator must be rather tolerant to outliers; thus, C is chosen quite large (typically, equal to the largest absolute temporal intensity difference). Then, as the accuracy of the dominant motion estimate improves, outliers are better identified and should be rejected. Therefore, C is lowered at each computation of a new increment $\Delta\Theta_k$, using the following schedule: $C_k = 0.9 \times C_{k-1}$, until it reaches a preset value C_p, or a robustly estimated value C_r.

Since the residual r'_i can be assimilated to a displaced frame difference, C can be viewed as equivalent to an intensity variation. Usually, an error of two to five grey levels is considered as acceptable in a matching issue or in a motion-compensated coding scheme for instance. Since C represents the value beyond which the contribution of the point to the estimation process becomes null, we will consider values of C_p ranging from 5 to 20. In fact, the choice of this parameter value mainly depends on the suitability of the motion model to the real motion: if the model fits closely to the true motion, C_p can be small ; otherwise, it is preferable to set C_p to a rather large value.

On the other hand, we could also estimate on-line the variance of the noise (for the data conform with the estimated model). Since our data contain outliers, the median absolute deviation (MAD) [25] should be used to estimate the standard deviation of the noise. It is given by:

$$\widehat{\sigma} = 1.48 \times \ \mathrm{Med}_i(|r_i - \mathrm{Med}_j(r_j)|) \tag{25}$$

This quantity can in fact be related to the constant C. For instance, in [17], it is recommended to use a proportionality factor of 4.7 between C and $\hat{\sigma}$ to ensure a better efficiency in case of Gaussian noise. Estimating $\widehat{\sigma}$ at each incremental step would be too costly. Thus, in order to evaluate C_r, we have used the following technique. After the last incremental estimation step performed at the coarsest level, expression (25) is used to compute $\hat{\sigma}$. Then, C_r is set to $4.7\widehat{\sigma}$. Experiments carried out with a given preset C_p value equal to 8 or the robustly estimated value C_r (varying from 4

to 15 depending on the experiments) gave similar results, but obviously at a higher computation cost in the second case. Moreover, these experiments have shown that the final value of C is not really a critical matter.

The algorithm we have described is summarized in Fig. 2. It will be called Robust MultiResolution algorithm (RMR) in the following. The threshold λ appearing in the stopping test is usually set to a small value (equal to 4 or 5), and few iterations are needed to reach convergence at each IRLS step.

$C \longleftarrow Max(|I_t^{L-1}|)$
$\widehat{\Theta^{L-1}} \longleftarrow$ IRLS (residual (5),C)
FOR l = level $L-1$ to level 0 DO
  iter $\longleftarrow$ 0
  DO
    $C \longleftarrow f(C)$
    $\widehat{\Delta\Theta^l} \longleftarrow$ IRLS (residual(10), C)
    $\widehat{\Theta^l} \longleftarrow \widehat{\Theta^l} + \widehat{\Delta\Theta^l}$ and iter $\longleftarrow$ iter +1
  WHILE (iter $< \lambda$ and $\|\widehat{\Delta\Theta^l}\| > \frac{d}{2^l}$)
  if $l \neq 0$: $\widehat{A^{l-1}} \longleftarrow P(\widehat{A^l})$, $\widehat{\xi^{l-1}} \longleftarrow \widehat{\xi^l}$
END FOR
$\widehat{\Theta_{est}} \longleftarrow \widehat{\Theta^0}$

Figure 2: Robust multiresolution algorithm (RMR)

We have established an alternative version of this method. It relies on the direct application of the IRLS procedure to the function E_r (relation (23). Here, contrary to the RMR algorithm where IRLS procedures are successively applied within the multiresolution framework, the minimisation of E_r is first reconverted into an equivalent weighted least-squares problem, and then, the latter is solved using a multiresolution scheme. More precisely, we have:

$$E_r(\Theta) = \sum_{X_i \in W} \rho\,(\,\mathrm{DFD}_\Theta(X_i)\,,\,C\,) = \sum_{X_i \in W} w_i\,(\,\mathrm{DFD}_\Theta(X_i)\,)^2 \qquad (26)$$

where $\mathrm{DFD}_\Theta(X_i)$ is given in (8). At the beginning, no estimate is available. Thus, all weigths are set to 1, and $E_r(\Theta)$ can be minimized using the MRLS procedure. We get an estimate $\hat{\Theta}$, which allows us to compute the weights w_i (at the finest resolution level) according to relation (22), in which the residual is the quantity $\mathrm{DFD}_{\hat{\Theta}}(X_i)$. Those weights are then "propagated" throughout the pyramid. To this end, we use the same Gaussian filtering and subsampling technique as the one used to build the image pyramid. With those precomputed weights, a new coarse-to-fine multiresolution estimation is performed; however, instead of using least squares at

each incremental estimation step like in MRLS, we utilize weighted least squares. This version (see Figure 3) avoids the computation of weights at each incremental estimation, but several passes through the pyramid are required. We will call this second method the Pseudo M-estimator (PSM) algorithm in the subsequent.

$w_i \longleftarrow 1.0, \quad k \longleftarrow 0 \quad \text{and} \quad \widehat{\Theta} \longleftarrow 0$
DO
$\widehat{\Theta}_p \longleftarrow \widehat{\Theta}$
$\widehat{\Theta} \longleftarrow \text{MRLS}(w_i)$
$w_i \longleftarrow \frac{\psi(\text{DFD}_{\widehat{\Theta}}(X_i), C)}{\text{DFD}_{\widehat{\Theta}}(X_i)}$ et $k \longleftarrow k+1$
WHILE $\|\widehat{\Theta} - \widehat{\Theta}_p\| > d$ and k$< \lambda_1$
$\widehat{\Theta_{est}} \longleftarrow \widehat{\Theta}$

Figure 3: Pseudo-M-estimator algorithm (PSM)

4.3 Complementary stages

Let us point out undesirable behaviours of the robust estimation algorithms described in the preceding section which may sometimes happen. They originate from three general problems related to the considered modeling :

1. the real motion that we aim at recovering, could be described by less parameters than those corresponding to the chosen model to be estimated ; equivalently, it may happen that the real underlying motion and the spatial distribution of intensity gradients do not sufficiently constrain the parameters of the considered model.

2. the initial minimization step of our algorithms are based on least-squares (this step is more significant in the PSM than in the RMR); therefore, the secondary objects and their motion could be "absorbed" in the minimization to sufficiently constrain the "degrees of freedom" left indeterminate (if any) by the observations corresponding to the dominant motion. This is further facilitated when the dominant motion areas are poorly textured.

3. since our algorithms are iterative, they depend on the initial guess ; if those initial estimates are quite far from the true parameter values, the estimation process may fall in an unsatisfactory local minimum.

A typical example is given in Figure 6. Figure 6.a displays the true flow vectors, and Figure 6.b shows the flow vectors corresponding to the estimated model that we suppose to have obtained (vectors are plotted only at positions where intensity

gradient is supposed to be important enough). The velocity vectors are roughly well estimated, although the two independent translations are blended by the estimation process into a single rotational motion that has nothing to do with the real motions. One way of alleviating these difficulties could be to use another estimator, like the least median of squares, at least for the first iteration. However, this solution suffers from other shortcomings, as pointed out in Section 4.1. In fact, an easier and efficient answer to this problem is to add the following stage to the RMR algorithms described in the previous section. We start the estimation process by considering a constant motion model from the coarsest resolution level $L-1$ to the level Lc (included). Then, from level Lc (also included) to the finest resolution, we continue the estimation with the more complex considered model (e.g., an affine one). This modified algorithm is called RMRmod.

Usually, the value of Lc is 2, whereas the value of L is 3 or 4. Let us note that in a sequence, it could be possible to use former estimates of the parameter vector Θ obtained in the previous frames to set an adequate value of Lc. The main advantage of this method is that the incremental estimation algorithm is very efficient when a constant model is used, as explained in [11]. Then, it helps to discriminate between several objects or between the background and objects.

The PSM algorithm can also be modified in that way, by estimating a constant model at the very beginning with the MRLS algorithm. This algorithm (named PSMmod) appears then similar to the scheme presented in [20]. However, in [20], after each multiresolution estimation step, an explicit detection algorithm (more complex than a simple weighting procedure) is used. It discards points for which the computed motion is not satisfactory, and attributes a weight equal to one to all other points. This might be a too "hard" decision rule if there is no relevant explicit detection map. Besides, after the constant motion model estimation stage, this decision rule can also lead to the selection of only a limited set of points, which may form a region which is likely to really undergo a translational motion. Subsequent estimations with an affine motion model will then be confined to recover the nearly constant motion of that small region. Thus, it could be not so easy to cope with situations involving more complex motions. In our case, the computation of the weights after the estimation of the constant motion model is performed with a constant C in the ψ function twice as large as the one used in the subsequent steps[2]. This is to take into account the fact that the model we have used may be a too simple one. Thus, we will keep more points in order to adaquately perform the estimation of all the affine parameters in the subsequent multiresolution estimations.

Finally, let us note that in the two methods, a point with a null weight at a given iteration is not definitively discarded, and can get a non zero weight in subsequent

[2]More generaly, in the PSM algorithm, modified or not, we could have change the C value according to a scheme similar to the one used in the RMR. However, some experiments performed with such a strategy showed that the impact on the results was far less significant, and even led sometimes to less satisfactory results than those obtained with the described algorithm.

iterations. Therefore, the two methods do not prevent us from estimating motions which are far from being constant, as shown in the next section.

5 Results

We have tested our algorithms on real images with synthetic motions, to quantitatively evaluate the results and to assess the performance of the algorithms. We have also carried out experiments with several real image sequences of quite different nature, including outdoor and indoor scenes, meteorological image sequences as well as underwater video sequences [21]. However, for lack of space, we will only report results obtained with two representative outdoor image sequences.

5.1 Experiments with real images and synthetic motion

The quantitative evaluation of our algorithms has been conducted on a set of N_{exp} experiments. For each experiment, we have applied a synthetic motion field to the real image of Fig. 12.a in order to construct a second image (a bilinear interpolation is used to determine the intensity values when the flow vector does not point to an element of the image grid). This motion field is composed of two different affine models. The first one is applied to a square window situated just below the middle of the image (area denoted Z_1, see Fig. 7.a), the second one to the rest of the image (area Z_2). The experiments differ from each other according to the affine models involved, which are in fact randomly selected. More precisely, the constant and linear coefficients are chosen at random using a uniform law in the intervals $[-3, 3]$ and $[-0.05\,, 0.05]$ respectively (the length unit is the pixel). The reference point of the models is the center of the square Z_1. Fig 7.b displays one of the synthetic fields generated that way. In practice, displacement magnitudes are ranging from 0 to about 15 pixels.

Each experiment consists in studying the behaviour of the estimators when varying the proportion of each area (Z_1 and Z_2) in the estimation support window. We proceed as follows. We estimate the six parameters of an affine motion model in a square window W of varying size S_W (see Fig. 7a). We consider as error measurements the following expression:

$$err_n(t_n) = \frac{\sum_{X_i \in (W \cap Z_n)} \|\vec{V}_{\hat{A}}(X_i) - \vec{V}_{A_n}(X_i)\|}{\sum_{X_i \in (W \cap Z_n)} \|\vec{V}_{A_2}(X_i) - \vec{V}_{A_1}(X_i)\|} \quad n = 1, 2 \tag{27}$$

where A_1 (resp. A_2) represents the first (resp. the second) motion model, $\hat{A}$ the estimated model, and t_n is the proportion of the surface of the computation window W covered by area Z_n, that is:

$$t_n = \frac{\#(W \cap Z_n)}{\#(W)} \tag{28}$$

Algorithms	L	λ	d	C_p	Nber of iterations in IRLS	Lc	λ_1
MRLS	4	6	0.1	-	-	-	-
RMR and RMRmod	4	6	0.1	8.0	4	2	-
PSM and PSMmod	4	6	0.1	8.0	-	-	4

Table 1: Values of the parameters for the different algorithms. The number N_{exp} of experiments is 150.

where # stands for the number of points in the considered region. Thus, when the estimated motion $\hat{A}$ corresponds to A_n, the value of the error err_n will be close to 0. The denominator term in the error expression has been introduced in order that the error would be near 1 when the estimated motion corresponds to the other model. Indeed, we have a bounded normalized error measurement. Otherwise, its value would have depend on the window size and on the rate of "similarity" between the two fields generated by the two affine models. Hence, it would not have been possible to compare the different experiments. The parameters used in each algorithm are given in Table 1.

Results are plotted in Fig. 8. We have drawn for each considered ratio t_n the average error value $\overline{err}_n(t_n)$ derived from the N_{exp} experiments. Let us note first that, because of the particular expression of the error function, curves $\overline{err}_1(t_1)$ (Fig. 8a) and $\overline{err}_2(t_2)$ (Fig. 8b) are nearly symmetrical with respect to the first bisecting line. From now, we will only consider $\overline{err}_1(t_1)$. As expected, the MRLS estimator averages the two motions (Fig. 9a) whereas the robust estimators provide correct estimations (Fig. 9b-e). In Fig. 8a, we can observe that the RMR performs better than PSM, especially when area Z_1 remains predominant in the estimation support. Let us consider that the estimated affine motion model does not correspond to any of the two motions A_1 or A_2 when the average error value lies between 0.1 and 0.9, and let us call "transition gap" l_{tg} the length of the interval of corresponding values of t_1 (i.e. such that $\overline{err}_1(t_1) \in [0.1, 0.9]$). This gap length is 0.6 for the MRLS, 0.41 for the modified PSM, 0.33 for PSM, and 0.24 for both versions of the RMR algorithm. If we consider now the standard deviation of the errors in the set of experiments, we can notice that those corresponding to the RMR algorithm are higher in the transition gap than the PSM ones. The RMR and modified RMR algorithms show a higher variability in their behaviour. In fact, within the transition gap and for a given value of t_1, they do not always estimate the same model depending on the experiment.

Besides, in our experiments, in which the linear part of the motion models is quite important, we can observe that the modification that we have added to the initial version of both RMR and PSM algorithms does not provide a real noticeable improvement. We can even see that the modified PSM algorithm gives rather bad results when area Z_2 is dominant in the estimation support W. This can be explained by looking at the synthetic flow field shown in Fig. 7b, which corresponds to one of

the experiments. The motion field in the area Z_2 is not translational at all. Thus, taking into account a constant motion model until the finest resolution level allows the estimator to retain only one part of the flow field, even after the introduction of the affine model in the subsequent iterations. On the contrary, this effect does not exist for the modified RMR algorithm, where the constant model is introduced only at the lowest resolution level.

We have also studied the performance of our algorithms in presence of noise on the intensity values. To this end, we have added to the constructed image in each experiment a zero-mean Gaussian white noise of standard deviation σ_G. Fig. 10a presents the obtained results for a significant noise level ($\sigma_G = 11$). While PSM and modified PSM algorithms do not really provide better estimates than MRLS, (see Fig. 10a and 10b), RMR and modified RMR algorithms remain rather insensitive to noise (Fig. 10c and 10d). This can be explained in part by the fact that the identification of the dominant motion and the rejection of outliers starts at the lowest resolution already, where the noise is filtered. This is obviously not the case for the PSM algorithm.

Let us point out that it can be not so easy to foretell for a given estimation window, which is or which should be the dominant motion. In fact, what counts is not merely the size of each region, but the number of points really conveying perceptible motion information. Since the motion computation relies on the motion constraint equation, where the spatial gradient of the intensity plays a crucial role, uniform areas (which may occupy a non negligible area in the example we deal with) deliver in fact no motion information. Thus, the transition point where we skeep from the estimation of the motion of area Z_1, to that of area Z_2, would better correspond to a window W such that:

$$\sum_{X_i \in W \cap Z_1} \|\vec{\nabla} I(X_i)\| = \sum_{X_i \in W \cap Z_2} \|\vec{\nabla} I(X_i)\| \tag{29}$$

Figures 11a et 11b show results obtained for two different locations of the estimation window W in the image of Fig. 12.a. In both cases, the synthetic flow field is still the one shown in Fig. 7b. In the first experiment, the window W is taken in the middle of the image. In the second one, the window W is placed on the projections of the cars in the left part of the image. In the first case, area Z_1 contains a region of leaves of low contrast compared to the projections of the other objects in the foreground (i.e., cars, sign). The transition point corresponding to "equilibrium" (29) and computed at level 0 of the pyramid is reached for a value of t_1 equal to 0.58. In the second case, the spatial intensity gradients in area Z_1 are much more important. The motion of area Z_1 should remain dominant up to $t_1 = 0.42$, considering again the relation (29) computed at the finest resolution level (level 0). As we can see from the curves plotted in Fig. 11, the location of the transition point for the PSM algorithms agrees with those values, while this is not the case with the RMR algorithm, (in the second experiment especially). Let us point out however that in the latter case, the transition profiles are sharp. The different behaviour of

the RMR algorithms (w.r. to PSM) can be partially explained by the fact, that the single least-square minimisation performed initially occurs at the lowest resolution level. Therefore, this is the only time when the image gradients are all considered equivalently.

Finally, it is worth noticing that the points where the error $\overline{err}_1(t_1)$ falls below 0.9 (and conversely, raises above 0.1) in the results reported above may be surprising. Indeed, the theoretical breakdown point of the M-estimator is $\beta = \frac{1}{n+1}$, where n is the number of parameter to be estimated. Hence, the length of the transition gap should be greater than $1 - 2\beta$, that is, in our case, approximately 0.7. However, the Monte-Carlo study that we have carried out leads to quite different values. Even the MRLS algorithm (based on the least-squares estimator whose β value is 0) supplies a better value than 0.7. We believe that the following points partially explain this difference:

- first, as it is explained in the previous paragraph, the observations at each point do not provide the same "quantity" of information;
- second, it is important to point out that the value β of the breakdown point is usually obtained by considering specific outliers distributions (those which lead to the worse estimation results for a given percentage of data contamination), whose probability of occurence in real experimental data sets, is likely to be very low;
- third, the multiresolution and incremental estimation scheme helps the discrimination between motions, as the spectral analysis performed in [11] explains it.

For those reasons, the theoretical value of the breakdown point may not always reflect the robustness of a given algorithm to real "usual" situations encountered in a given application.

5.2 Experiments with real sequences

"Cars" sequence.

Figure 12.a presents the first image of the sequence "Cars". Figure 12.b shows the temporal image difference between the two considered images (to which an offset of 128 was added : a grey value of 128 therefore corresponds to a null difference, and the more black or white a point is, according to the sign of the difference, the greater the magnitude of the difference). Motion is composed of four main components : panning of the camera from the right to the left (producing as a first order approximation an opposite apparent translation in the image), sway of the foliage, especially in the middle, and displacements of the two cars. To illustrate several typical cases, the image was divided in four blocks. In each one, the dominant motion is estimated and the value of $\mathrm{DFD}_{\hat{\Theta}}$ is then computed using the estimated

Algorithms	L	λ	d	C_p	Nber of iterations in IRLS	Lc	λ_1
MRLS	4	5	0.1	-	-	-	-
RMRmod	4	5	0.1	8.0	8	2	-
PSMmod	4	7	0.1	8.0	-	-	8

Table 2: Values of the parameters in the different algorithms.

motion. We use four levels in the estimation pyramid, (three might be enough with respect to the motion magnitude; in fact, we use a simple criterion based on the size of the support region to choose the number of levels).

Figures 13 and 14 reports results obtained with the MRLS and modified PSM algorithms. In the two upper blocks of the image, the motion of the leaves is globally rather incoherent and more like a "Brownian motion". Therefore, the dominant motion is clearly the panning, and both MRLS and modified PSM algorithms successfully estimate it, as shown by the flow field displayed in Figure 14 (displacement vectors are plotted only at points where the dominant motion is considered as appropriate according to the value of weight w_i). However, in the lower blocks, the rigid motions of the two cars are coherent. Figure 13.a as well as the estimated velocity field (Fig.14.a) clearly confirm the averaging effect of the MRLS algorithm : the DFD is neither correct in the static part of the scene nor on the projections of the cars; the motions are also badly estimated. On the contrary, the modified PSM algorithm perfectly computes the panning motion as shown in Fig. 13.b and Fig.14.b. Not only the panning motion is well compensated for but also the moving areas are more accentuated than using the MRLS. This is of particular interest if a subsequent detection stage is considered to delineate moving objects in the scene. Of course, it is possible to consider a second step, in which we estimate the second dominant motion for the points considered as not belonging to the first dominant motion, and so on.

"Roundabout" sequence.

Figure 15 contains three images from the sequence *roundabout* used in the second real experiment reported in this paper. Here, the camera is mounted on the left side of a car approaching a roundabout and is pointing laterally. The dominant motion in the sequence is therefore due to its movement, but, since significant differences in depth occur in the scene, the 2D dominant motion model corresponds to the motion of the background only (i.e., houses mainly).

In this experiment, we have estimated an affine motion model between each pair of consecutive frames, using MRLS, modified PSM and modified RMR algorithms. The value of the different parameters involved in each algorithm are given in Table 2. These computed motion models serve to produce the motion-compensated image sequences of Figures 16, 18 and 17. If a region is well compensated for using the estimated motion, it should stay at the same location in successive images. Once

more, Figure 16 clearly indicates the averaging effect of the MRLS algorithm, since the whole image "loses its shape" over time. The modified PSM algorithm delivers the background motion until frame 63, but averages both background and sign motions from frame 63 to 67 (see in Fig. 17 : the houses appear no longer fixed and are translating to the right ; the sign moves to the left). Finally, the modified PSM algorithm "sticks" on the sign motion (although the sign is not completely well registered). On the contrary, Figure 18 shows that the modified RMR algorithm coherently estimates the background apparent motion along the entire sequence : the houses in the background clearly remains static in the motion-compensated sequence. The differences in the performances of the two robust algorithms will be commented upon in the next section from a more general point of view.

5.3 Comparison between the modified RMR and modified PSM algorithms

As presented above, these two algorithms often yield similar results, especially when there is an obviously dominant area where the considered motion model can fit the underlying optical flow. However, several differences explaining the results obtained in the synthetic experiments as well as with the *roundabout* sequence, can be identified :

1. assuming a constant model at the coarsest level only is a weaker hypothesis than assuming a constant model for a first complete multiresolution estimation step ; this implies that the modified RMR algorithm can recover a broader class of affine motion than the modified PSM algorithm.

2. in the PSM algorithm, all points are used equivalently in the first multiresolution estimation step, which tends to produce an averaged initial solution. On the contrary, the RMR algorithm discards significant outliers from the very beginning in the multiresolution estimation.

Let us consider now the computational aspect. The PSM algorithm requires several passes through the pyramid. Its time complexity is about the time complexity of the MRLS algorithm times the number of passes. In the RMR algorithm, there is only one coarse-to-fine estimation step, but the estimation of each increment $\Delta\Theta_k$ induces a minimization based on the IRLS procedure. In fact, Table 3 shows (for our implementation) that RMR and MRLS time complexity are equivalent. This can be explained as follows. On one hand, IRLS usually converges very rapidly, and thus the IRLS procedure in fact does not consume too much time; on the other hand, the number of increments to be computed at one given level to reach stability is less important for the RMR algorithm than for the MRLS and PSM algorithms because of the IRLS performance (i.e. the increments are better estimated). Then, since each incremental computation implies intensity interpolations for every point (here, we perform bilinear interpolation), the RMR algorithm saves cpu time here compared to the two other algorithms.

Algorithm	MRLS	RMRmod	PSMmod
Sparc 2	13.33	13.55	43.66
Sparc 10	4.63	4.63	14.40

Table 3: average cpu time (in seconds) of the estimation. (the ten 256×256 images from the *roundabout* sequence are considered). The number of incremental calculations at one level is always limited to 8 ; the number of iterations in IRLS is limited to 6, and the number of coarse-to-fine estimations is limited to 6 for PSM.

6 Conclusion

We have described in this paper two robust multiresolution algorithms to estimate parametric motion models. They have been favorably compared to a multiresolution least-mean-squares method. This has been validated on synthetic as well as real motion examples and on images depicting complex scenes. Such algorithms are of great importance, since they can evaluate the global motion in the image or over a region without being affected by secondary motions, and without requiring an explicit segmentation step. Indeed, they can be seen as an efficient first stage for the detections of moving objects in a sequence acquired by a mobile camera, as demonstrated in [31].

This study is supported in part by the French Ministry of Research in the context of the GDR-PRC "Man-Machine Communication" (Computer Vision Program, MRT contract 91S269), and by "Région Bretagne"(Brittany Council) through a contribution to student grant.

References

[1] J.K. Aggarwal and N. Nandhakumar. On the computation of motion from sequences of images- a review. *Proc. of the IEEE*, Vol.76, No.8:917–935, August 1988.

[2] B. Bascle and N. Deriche. Stereo matching, reconstruction and refinement of 3D curves using deformable contours. In *Proc. 4th Int. Conf. Computer Vision, Berlin*, pages 421–430, May 1993.

[3] R. Battiti, E. Amaldi, and C. Koch. Computing optical flow across multiple scales: an adaptative coarse-to-fine strategy. *Intern. J. Comput. Vis.*, 6:2:133–145, 1991.

[4] J.R. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. of* 2^{nd} *European Conf. Computer Vision*, pages 237–252, Springer-Verlag, S.Margherita Ligure, Italy, 1992.

[5] J.R. Bergen, P.J. Burt, R. Hingorani, and S. Peleg. Computing two motions from three frames. In *Proc. 3rd Int. Conf. on Computer Vision*, pages 27–32, Osaka, Dec. 1990.

[6] M. J. Black. *Robust incremental optical flow*. PhD thesis, N^o 923, Yale University, Computer Science Dept, September 1992.

[7] M.J. Black and P. Anandan. *The robust estimation of multiple motions: affine and piecewise-smooth flow fields.* Technical Report SPL-93-092, Xerox, Palo Alto Research Center, December 1993.

[8] A. Blake. Comparison of the efficiency of deterministic and stochastic algorithms for visual reconstruction. *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.11, No.1:2–12, Jan. 1989.

[9] P. Bouthemy and E. François. Motion segmentation and qualitative dynamic scene analysis from an image sequence. *Int. Journal of Computer Vision*, Vol.10, No 2:157–182, April 1993.

[10] P.J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pages 6–35, Springer-Verlag, 1984.

[11] P.J. Burt, R. Hingorani, and R.J. Kolczynski. Mechanisms for isolating component patterns in the sequential analysis of multiple motion. In *IEEE Workshop on Visual Motion*, pages 187–193, Princeton, October 1991.

[12] T. Darrell and A. Pentland. Robust estimation of a multi-layered motion representation. In *Proc. IEEE Workshop on Visual Motion*, pages 173–178, Princeton, Oct. 1991.

[13] W. Enkelmann. Investigations of multigrid algorithms for the estimation of optical flow fields in image sequences. *Computer Vision, Graphics and Image Processing*, Vol.43:150–177, 1988.

[14] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics : The Approach Based on Influence Functions.* John Wiley and Sons, New York, 1986.

[15] F. Heitz and P. Bouthemy. Multimodal estimation of discontinuous optical flow using Markov random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol 15(12):1217–1232, Dec. 1993.

[16] M. Hoetter. Differential estimation of the global motion parameters zoom and pan. *Signal Processing*, Vol.16:249–265, 1989.

[17] P.W. Holland and R.E. Welsch. Robust regression using iteratively reweighted least squares. *Commun. Stat.- Theor. Meth.*, A6:813–828, 1977.

[18] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, Vol.17:185–203, 1981.

[19] P.J. Hubert. *Robust statistics*. Wiley, 1981.

[20] M. Irani, B. Rousso, and S. Peleg. Detecting and tracking multiple moving objects using temporal integration. In *Proc. of 2nd ECCV-92, S.Margherita Ligure, Italy*, pages 282–287, Springer-Verlag, May 1992.

[21] Jean-Marc Odobez. *Estimation, détection et segmentation du mouvement dans une séquence d'images: une approche robuste et markovienne*. PhD thesis, University of Rennes I, N^o 1304, December 1994.

[22] J.M. Jolion, P. Meer, and S. Batauche. Robust clustering with application in computer vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13:791–802, August 1991.

[23] J. Konrad and E. Dubois. Multigrid Bayesian estimation of image motion fields using stochastic relaxation. In *Proc. 2nd Int. Conf. Computer Vision*, pages 354–362, Tarpon Springs, Florida, Dec. 1988.

[24] R. Kumar and A.R. Hanson. Robust estimation of camera location and orientation from noisy data with outliers. In *Proceedings IEEE Workshop on Interpretation of 3D Scenes*, pages 52–60, Austin, Texas, November 1989.

[25] P. Meer, D. Mintz, and A. Rosenfeld. Robust regression methods for computer vision: a review. *International Journal of Computer Vision*, 6(1):59–70, 1991.

[26] F.G. Meyer and P. Bouthemy. Region-based tracking using affine motion models in long image sequences. *CVGIP: Image Understanding*, Vol. 60(2):119–140, September 1994.

[27] A. Mitiche and P. Bouthemy. Computation and analysis of image motion: a synopsis of current problems and methods. *Intern. J. Comput. Vis.*, 1994. Accepted for publication.

[28] H.H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing Jal*, Vol.6, No.2:pp 59–74, May 1988.

[29] S. Negahdaripour and S. Lee. Motion recovery from image sequences using first-order optical flow information. In *Proc. of the IEEE Workshop on Visual Motion, Princeton*, pages 132–139, Oct. 1991.

[30] H. Nicolas and C. Labit. Global motion identification for image sequence analysis and coding. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 2825–2828, Toronto, May 1991.

[31] J-M. Odobez and P. Bouthemy. Detection of multiple moving objects using multiscale MRF with camera motion compensation. In *Proc.* 1^{st} *IEEE Int. Conf. Image Processing*, pages 257–261, Austin, Texas, November 1994.

[32] P.J. Rousseeuw. Least median of squares regression. *Jal of the American Statistical Association*, Vol.79, No 388:871–880, Dec. 1984.

[33] J. Schmetz and M.S. Mhita. Diurnal and interdiurnal variability of IR and WV brightness temperatures from Meteosat. *ESA Journal*, 13:329–341, 1989.

[34] W.B. Thompson, P. Lechleider, and E.R. Stuck. Detecting moving objects using the rigidity constraint. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2(15):162–166, February 1993.

[35] S. F. Wu and J. Kittler. A gradient-based method for general motion estimation and segmentation. *Jal of Visual Communication and Image Representation*, 4(1):25–38, March 1993.

[36] W.-Z. Zhao, F.-H. Qi, and T.Y. Young. Dynamic estimation of optical flow field using objective functions. *Image and Vision Computing*, Vol.7, No.4:259–267, Nov. 1989.

List of Figures

List of Tables

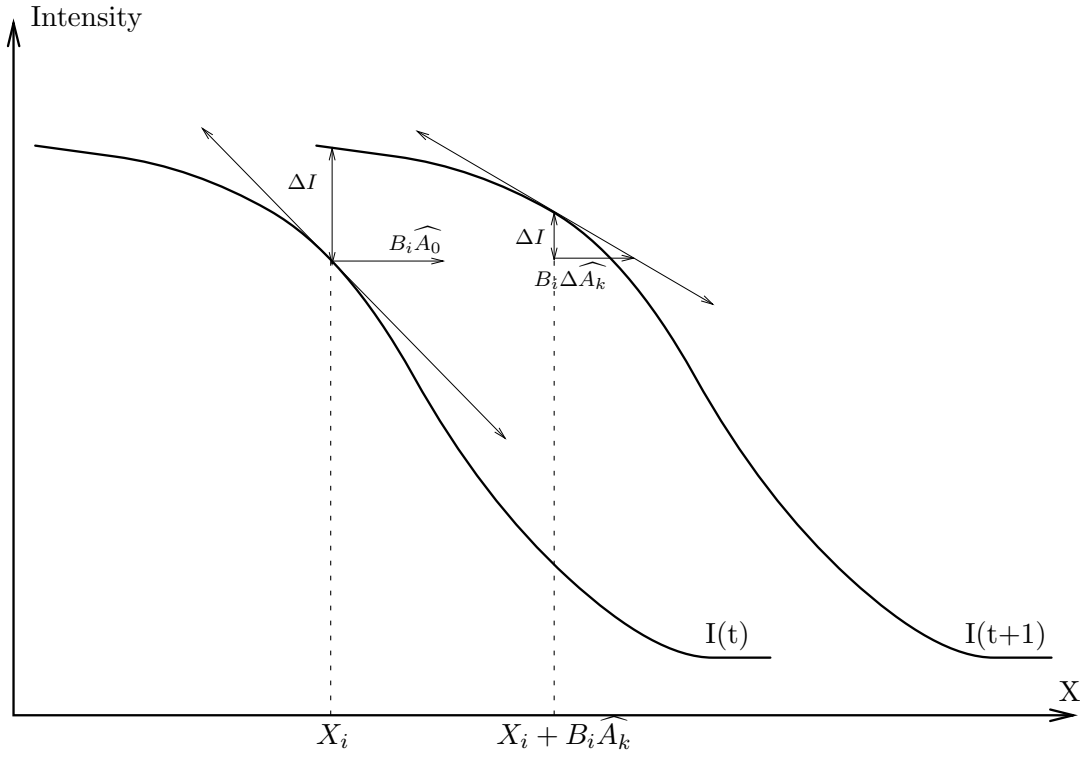


Figure 4: Geometric interpretation in one dimension of the incremental scheme.

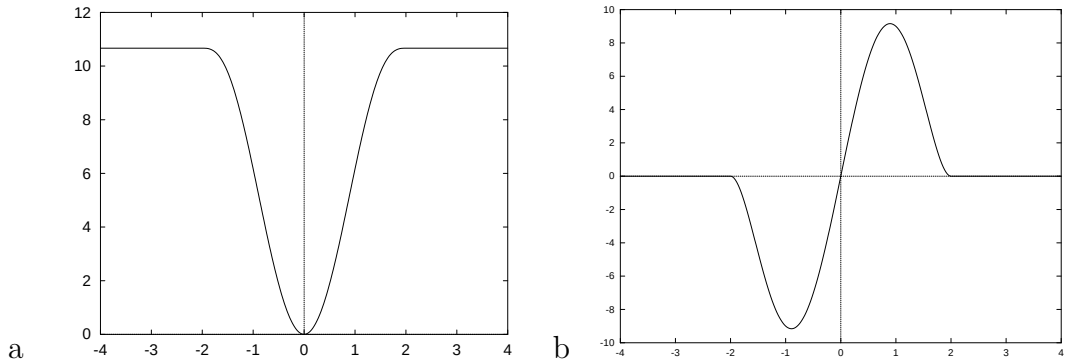
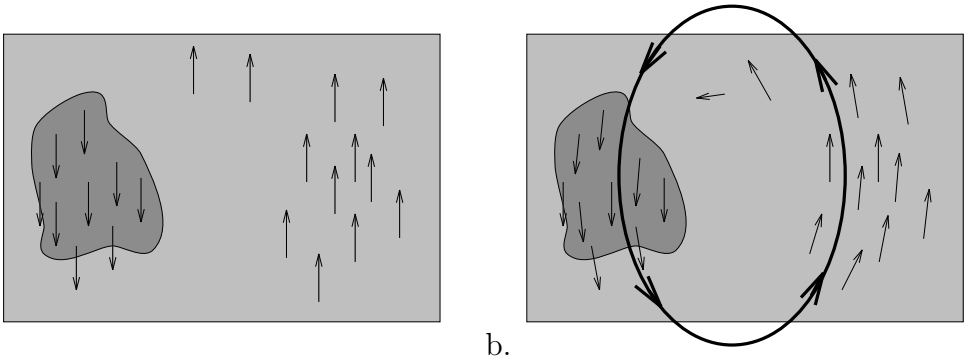
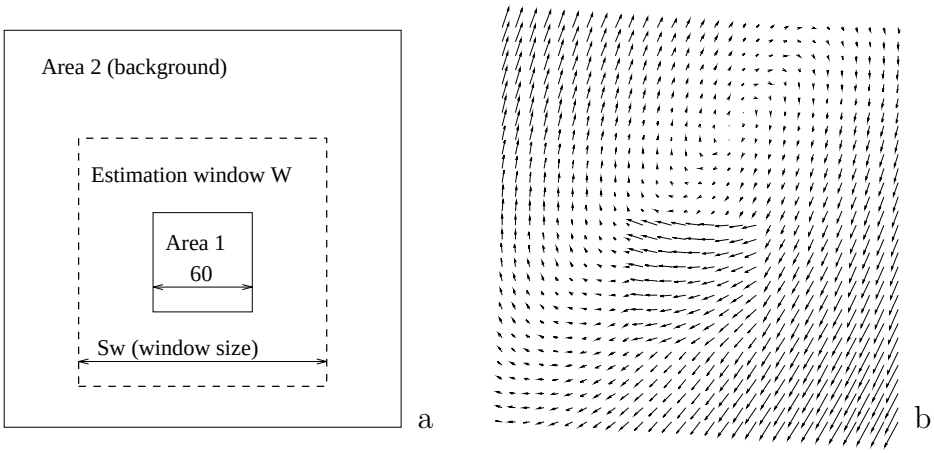


Figure 5: Tukey's "biweight" estimator. a) function ρ. b) Influence function ψ. $(C = 2)$.



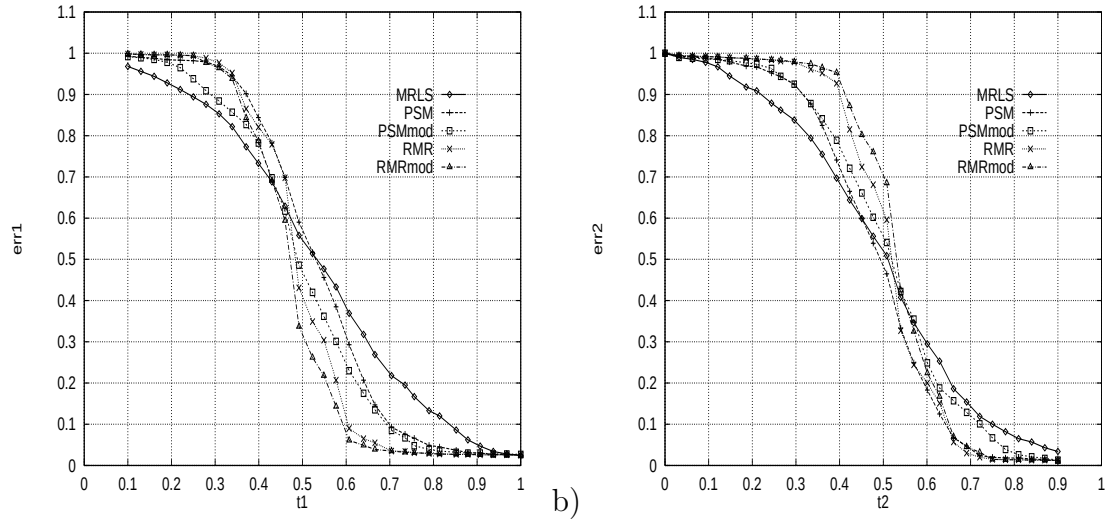
a. b.

Figure 6: a) Example of ambiguous problem : the dominant object translates upwards (flow vectors are plotted where significant intensity gradients are supposed to be located) and the second object translates downwards ; b) The resulting estimated field could be a rotational field at the cost of only a slight distorsion.



a b

Figure 7: a) experimental framework ; b) example of synthetic motion field used.



a) b)

Figure 8: a) and b) Average errors $\overline{err}_n$ obtained for each of the five algorithms as a function of the percentage t_n of window W occupied by area Z_n: a) $n = 1$; b) $n = 2$.

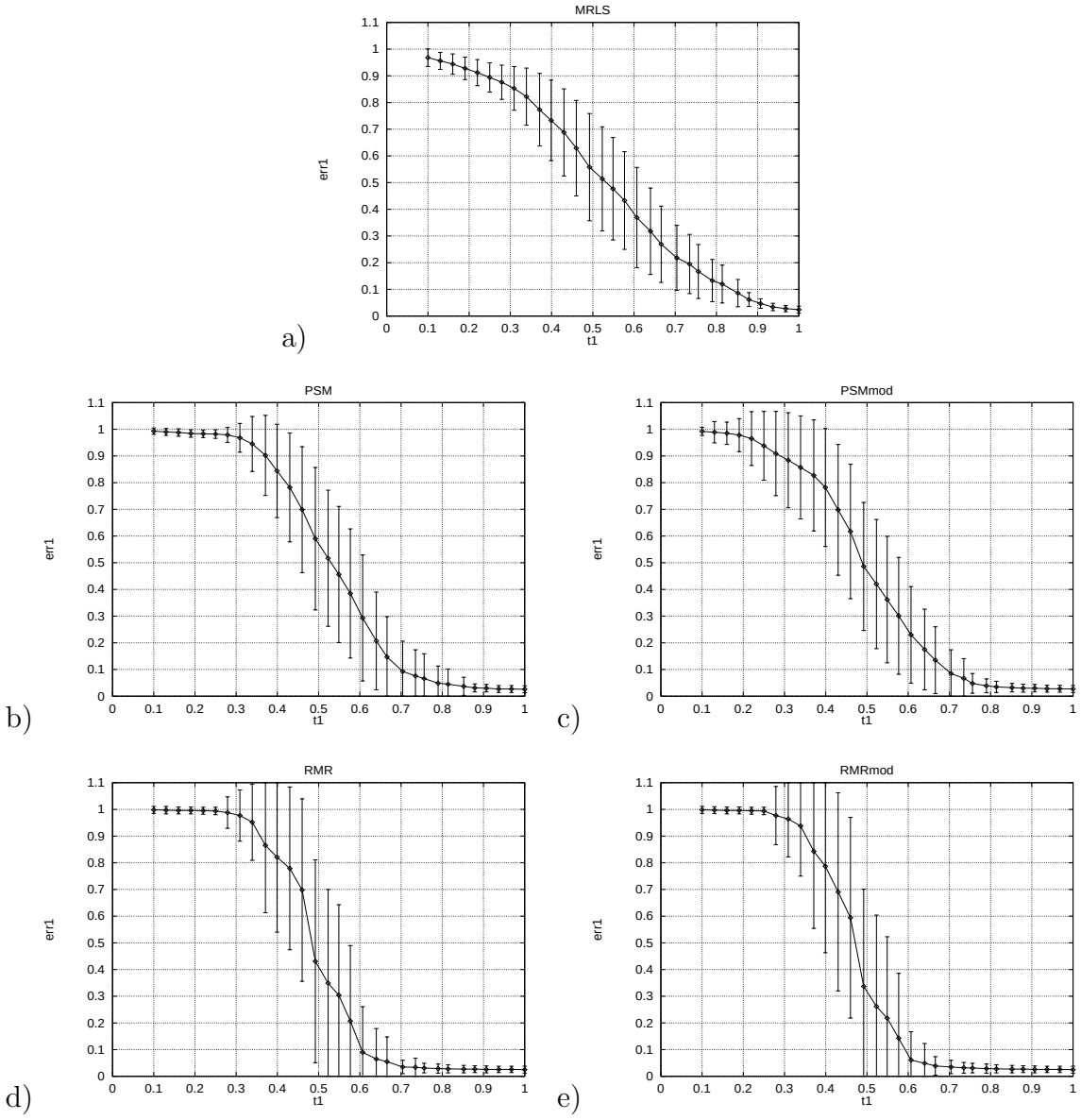


Figure 9: a)-e) Average error $\overline{err}_1$ and standard deviation σ_{err_1} as a function of t_1, obtained considering algorithm: a) MRLS, b) PSM c) PSMmod d) RMR et e) RMRmod.

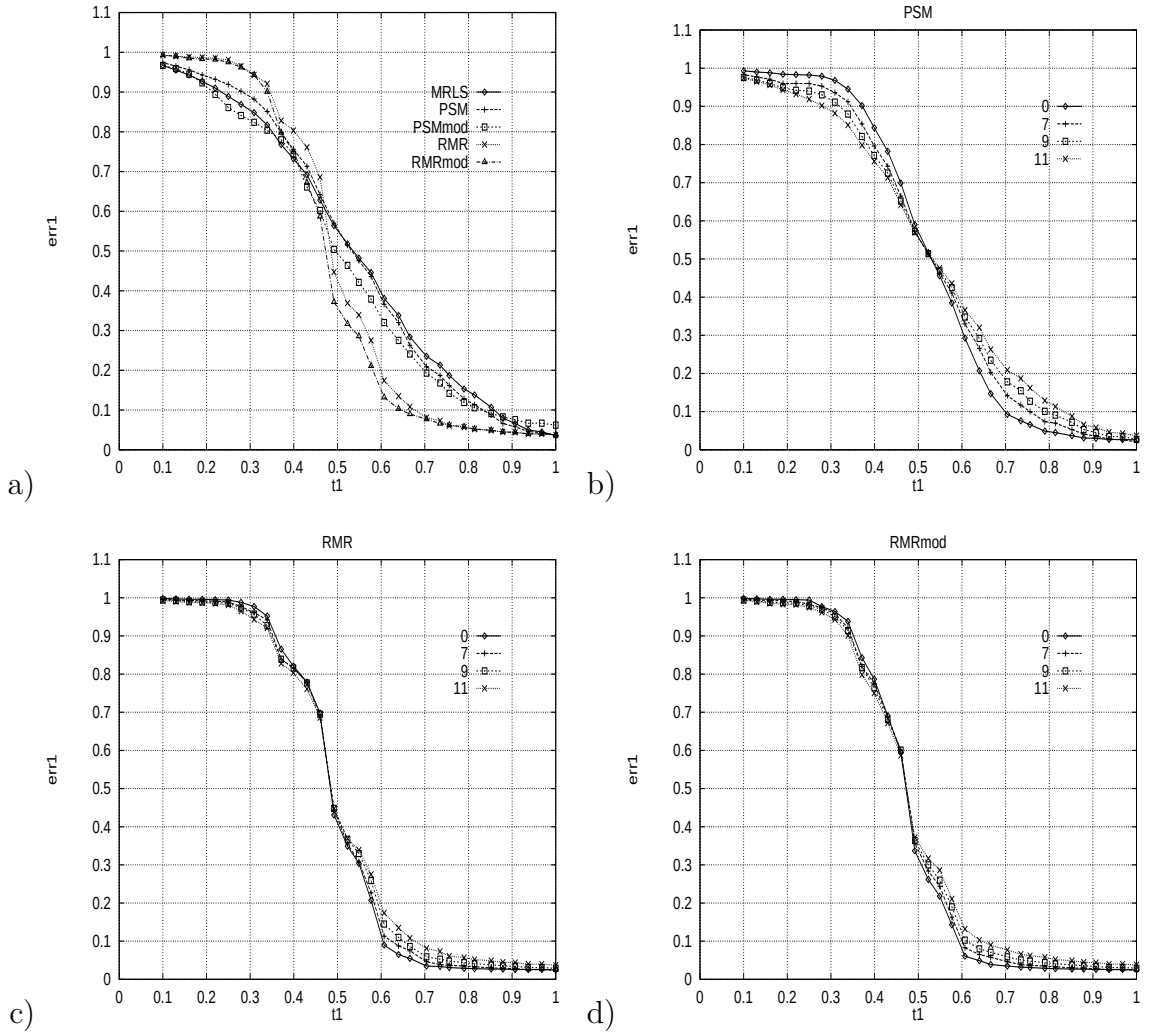


Figure 10: a) Average error $\overline{err}_1$ obtained for each of the five algorithms as a function of the percentage t_1 of the support window W occupied by area Z_1, with a zero-mean Gaussian white noise of standard deviation $\sigma_G = 11$ added to the constructed images. b) to d) Average error $\overline{err}_1$ as a function of t_1 for different values of the standard deviation σ_G of the Gaussian noise, for resp. the algorithm b) PSM, c) RMR and d) RMRmod.

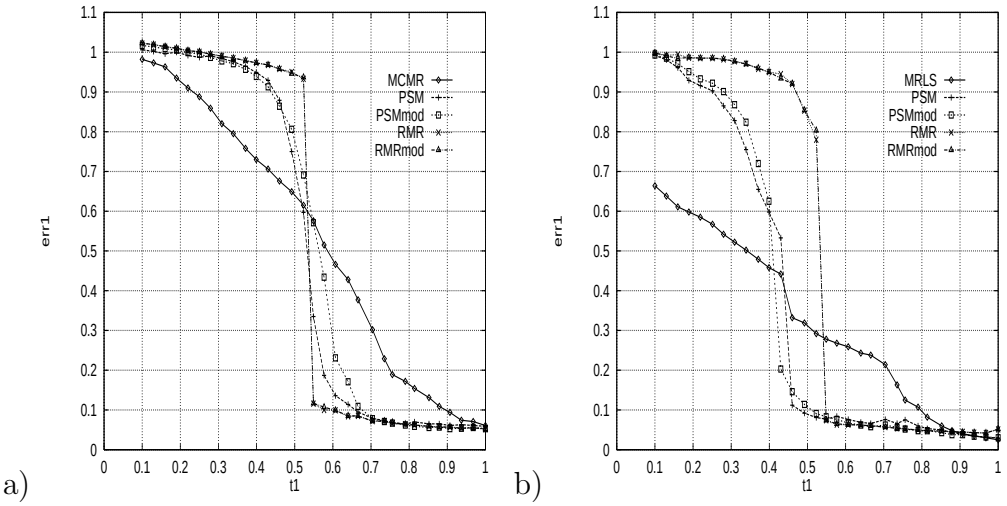


Figure 11: Average error $\overline{err}_1$ obtained for the particular experiment corresponding to the flow field of Figure 7b, when area Z_1 is located resp.: e) in the center of the first image of the "car" sequence; f) on the projections of the cars in the left of this image.

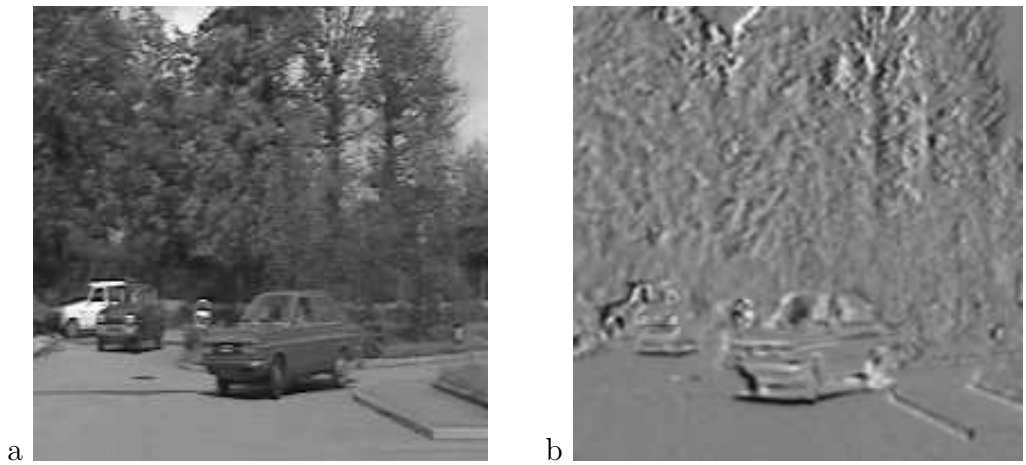


Figure 12: "cars sequence": a) first image ; b) temporal difference between the two considered images ;

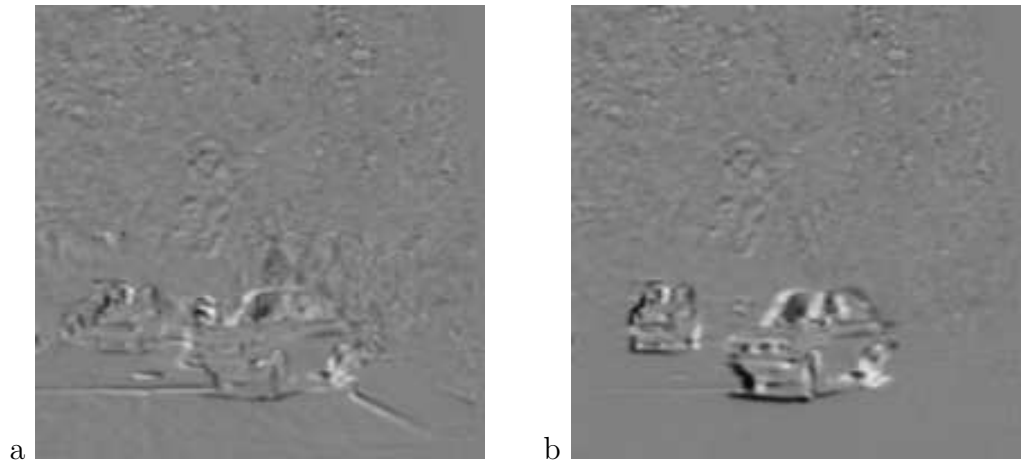


Figure 13: Compensated images DFD_{comp}, case of a velocity field estimated with : a) MRLS ($\lambda = 5, d = 0.1$); b) modified PSM ($C = 9, \lambda = 5, d = 0.1, \lambda_1 = 12$) ;

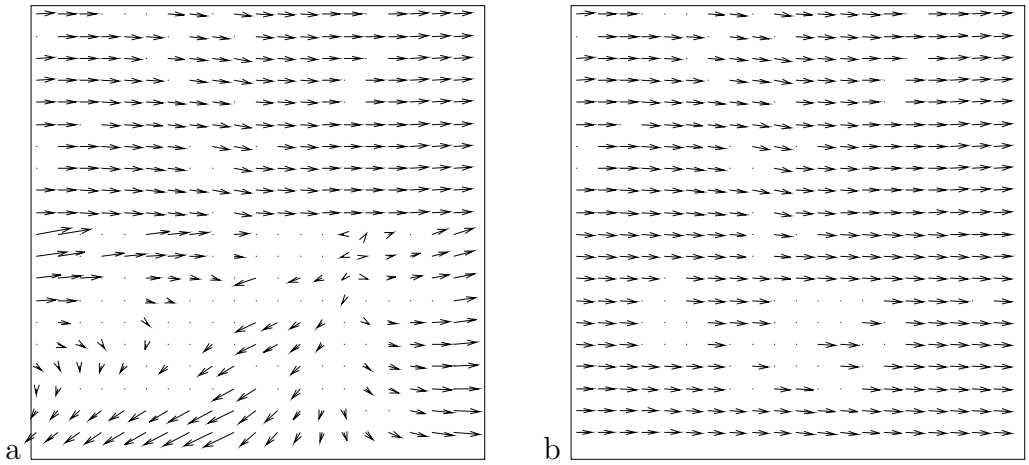


Figure 14: Velocity field corresponding to the model estimated with : a) MRLS ; b) modified PSM ($C = 9$) (For display convenience, velocity fields are subsampled by a factor of 9).

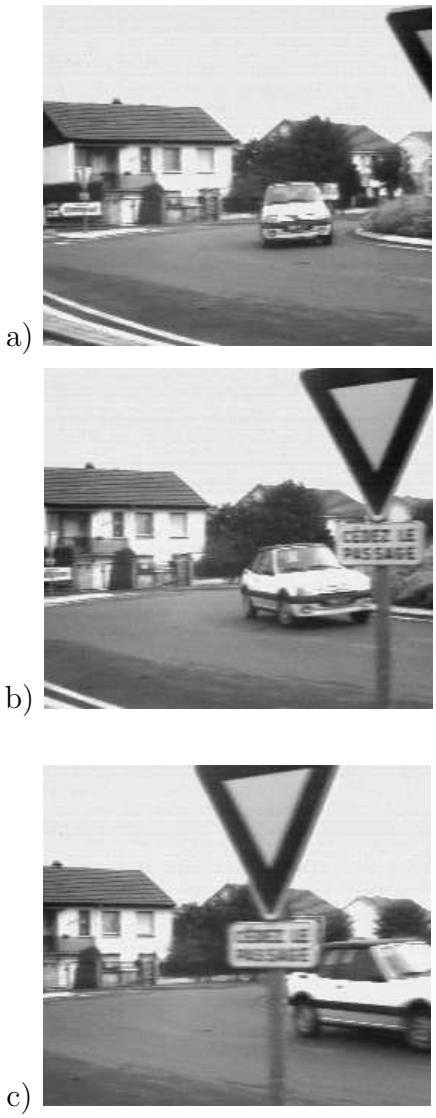


Figure 15: Three images out of sequence "roundabout" at time a) t_{62} b) t_{67} c) t_{72}.

Figure 16: Images at time t_{62}, t_{67} and t_{72} compensated with the motion computed with MRLS algorithm.

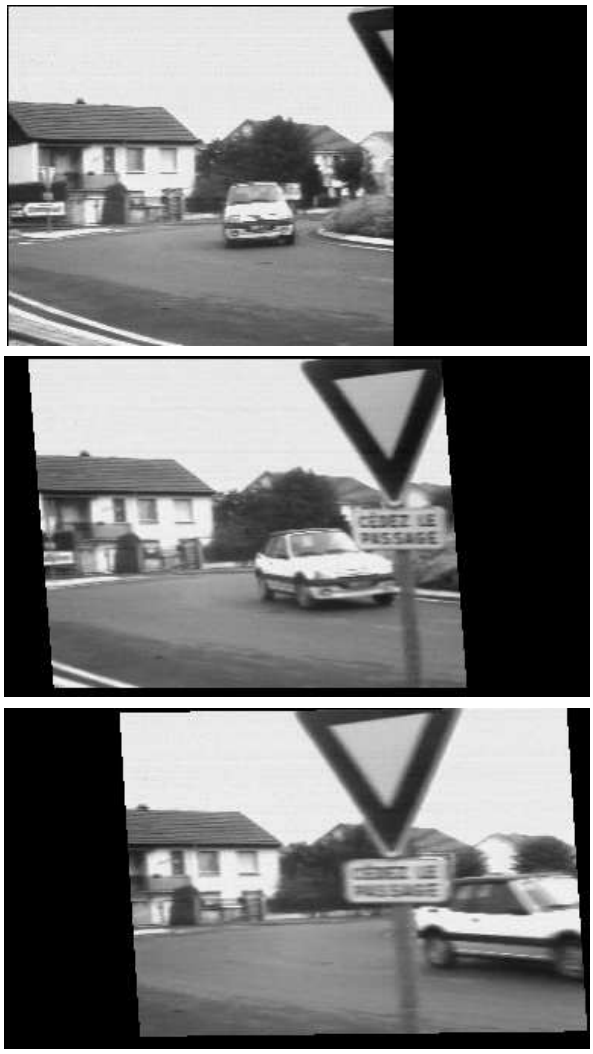


Figure 17: Images at time t_{62}, t_{67} and t_{72} compensated with the motion computed with the modified PSM algorithm.

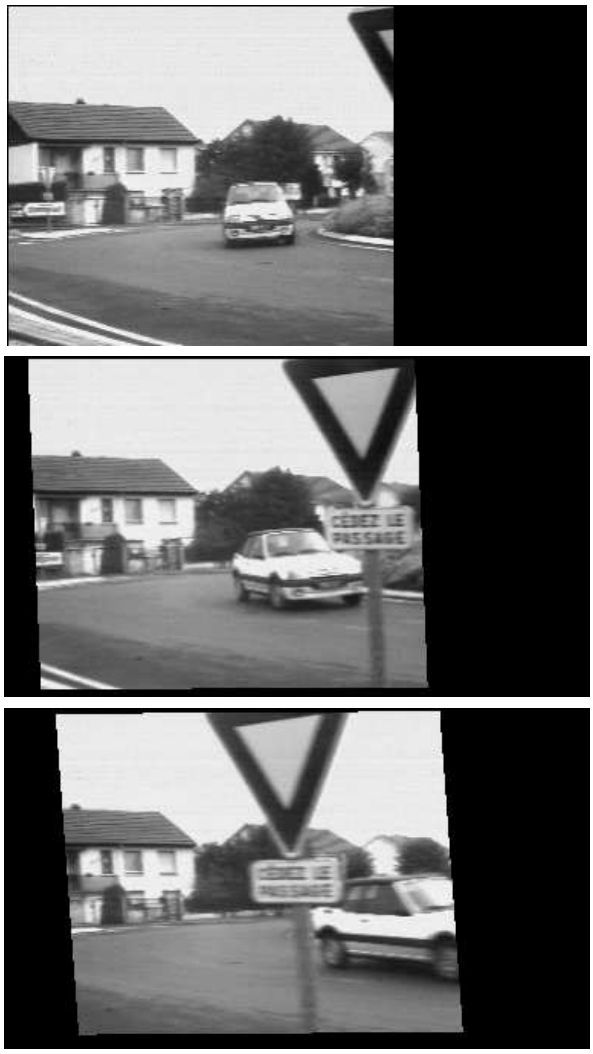


Figure 18: Images at time t_{62}, t_{67} and t_{72} compensated with the motion computed with the modified RMR algorithm.