Robust multiresolution estimation of parametric motion models applied to complex scenes

Article · January 1994

CITATIONS
50

READS
111

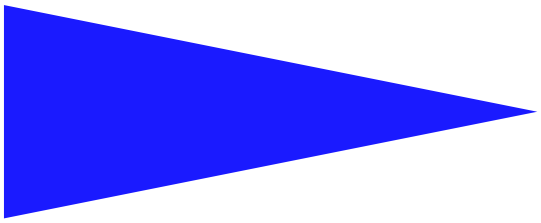
2 authors:

Jean-Marc Odobez
Idiap Research Institute
322 PUBLICATIONS **9,948** CITATIONS
SEE PROFILE

Patrick Bouthemy
National Institute for Research in Computer Science and Control
322 PUBLICATIONS **9,301** CITATIONS
SEE PROFILE

I R I S A
INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION
INTERNE
N° 788



ROBUST MULTIRESOLUTION ESTIMATION OF PARAMETRIC MOTION MODELS APPLIED TO COMPLEX SCENES

JEAN-MARC ODOBEZ AND PATRICK BOUTHEMY

ISSN 1166-8687

IRISA
CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 99 84 71 00 – Fax : (33) 99 38 38 32

Robust multiresolution estimation of parametric motion models applied to complex scenes

Jean-Marc Odobez*and Patrick Bouthemy*

Programme 4 — Robotique, image et vision
Projet Temis

Publication interne n ° 788 — Janvier 1994 — 30 pages

Abstract: This report describes a parametric motion model estimation algorithm. Motivations for the use of such models are, on one hand, their representativeness, which has proved to be efficient in numerous issues such as measurement, segmentation, tracking and interpretation of motion, on the other hand, their compactness and, therefore, the low computational cost of the estimation of these models. 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 developed two M-estimator-like techniques in a multiresolution framework. Numerical results support this approach, as demonstrated by the use of these algorithms on complex sequences.

Key-words: Computer vision, dynamic scene analysis, parametric models, motion estimation, robust estimation, multiresolution, dynamic segmentation.

(Résumé : tsvp)

*{bouthemy}{odobez}@irisa.fr

Centre National de la Recherche Scientifique
(URA 227) Université de Rennes 1 – Insa de Rennes

INRIA
Institut National de Recherche en Informatique
et en Automatique – unité de recherche de Rennes

Estimation robuste multi-échelle de modèles paramétrés de mouvement sur des scènes complexes

Résumé : Nous présentons dans ce rapport une méthode originale d'estimation de modèles paramétrés de mouvement. L'intérêt de formuler un problème d'analyse de mouvement par l'identification de tels modèles est double. Il s'agit d'une part d'une représentation compacte qui s'avère adéquate et pertinente dans des contextes différents et nombreux (mesure, segmentation, suivi ou caractérisation du mouvement). D'autre part, on peut en obtenir une estimation peu coûteuse en temps calcul. Le point crucial par contre est d'en obtenir une estimation fiable et précise, et de gérer correctement la présence de plusieurs mouvements dans l'image. Pour répondre de façon bien fondée et efficace à ces problèmes, nous avons développé un estimateur robuste, du type M-estimateur, défini dans un schéma multi-résolution. Cette technique a permis d'obtenir des résultats tout-à-fait satisfaisants sur des images représentant des scènes complexes.

Mots-clé : Séquence d'images, mouvement, modèles paramétrés, estimateur robuste, mesure multi-échelle, segmentation.

1 Introduction

One of the major areas in computer vision research is dynamic scene analysis, which has motivation from numerous applications ([AN88, Nag88, MB92c]). 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 complex 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 report, 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 [WK93, BF93], extraction and coding of temporal information in a motion-compensated coding scheme [Hoe89, NL91b], apparent motion estimation [ZQY89], tracking [MB92b], and recovery of useful 3D qualitative [BF93] or quantitative [NL91a] motion information. Similar models have been used with success for the registration of stereo images [BD93]. 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.

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 [NL91a]) whose efficiency usually deeply depends on the accuracy of these results. We present in this report two robust multiresolution algorithms for parametric motion models estimation.

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 [BAK91, Enk88, HB93, KD88]. Accurate estimations can be recovered even with large displacements or with an irregular intensity gradient distribution in the image. More recently, [BAHH92] and [MB92a] propose a multiresolution "least mean squares" motion parameters estimation technique. However, to be useful, the regions where we apply this technique have to be sufficiently large. Thus, if no segmentation is available, the *a priori* region where the estimation is performed (the whole image or blocks of big size) may contain several motions and the results will surely be affected. In [BAHH92], it is postulated that the global motion results from egomotion, and that the projected moving objects occupy only a very small part of the scene. There is less constraint in [BBHP90], where transparent motion is under consideration, but only constant models are used. In [MB92a], 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 computationnally heavy and difficult task. The class of robust estimators [Hub81, Rou84], which has become popular in image processing [MMR91, Bla92, JMB91] offers an appealing direction of investigation. Such an approach has already been used in [DP91] for motion estimation, but only for a 3D translation with constant depth ; i.e., a second order 2D motion model with three parameters at a single resolution has been considered. However, 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 report is organized as follows : in the following section, we describe our motion model and the objective function used in the minimisation. Section 3 presents the multiresolution least-mean-squares estimation algorithm, to which our new methods will be compared. Section 4 makes a brief review of robust estimators, and describes our solutions. Finally we will show results with simulated and real motion.

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 at point X_i ; B is a matrix, whose form depends on the chosen model, but where its 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 flow 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 [BF93, NL91a].
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. $S(X_i,t) = 0$ [HS81], 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, [SM89]). Hence, to deal with those changes, we choose S constant over the given region, that is:

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

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):

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

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

3 Multiresolution least mean squares estimation

We briefly describe here the multiresolution least mean square estimation scheme. It was introduced in [MB92a], and leads to an incremental estimation of the motion model through coarse-to-fine refinement. We added at each resolution level an incremental estimation step which improves results previously obtained in [MB92a]. In addition, we introduced the parameter ξ presented in the previous section.

3.1 Incremental estimation

The following presentation concerns the incremental step at one resolution level as well as from one level to the next. The principle of this incremental scheme is presented in the figure 1 in the monodimensional case. As described in the former section, equation (4) results from the assumption $\frac{dI}{dt}(X_i,t) = -\xi$. The following residual may also be derived from this assumption, if we approximate the derivative by a finite difference :

$$r'_i = I(X_i + \delta X_i,\ t+\delta t) - I(X_i,t) + \xi\delta t, \tag{6}$$

where $\delta X_i = V(X_i)\delta t$, and δt is the time step between two successive frames. We therefore want to minimize the following sum of squared difference errors (from right here, we take $\delta = 1$ to simplify notations) :

$$E_{\mathcal{A}}(\Theta) = \sum_{X_i \in F} (r'_i)^2 = \sum_{X_i \in F} (I(X_i + B(X_i)A,\ t+1) - I(X_i,t) + \xi)^2 \tag{7}$$

where F denotes the region (called the estimation support region) over which minimization is performed (e.g., the whole image or a block) and where we have substituted $V_A(X_i)$ for $V(X_i)$. To achieve minimization, we make use of an incremental algorithm. Let $\widehat{\Theta}^t_k = (\widehat{A}^t_k, \widehat{\xi}_k)$ 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 therefore:

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

A first order expansion of I around point $X_i + B_i\widehat{A}_k$ at time $t+1$ is performed for each residual r'_i, leading to the new error quantity :

$$\begin{array}{rcl} r''_i & = & I(X_i + B_i\widehat{A}_k, t+1) - I(X_i,t) + \widehat{\xi}_k \\ & & +\nabla I^t(X_i + B_i\widehat{A}_k, t+1)\, B_i\Delta A_k + \Delta\xi_k \\ & = & \mathcal{X}'_i\, \Delta\Theta_k - \mathcal{Y}'_i \end{array} \tag{8}$$

$$\text{where } \left\{ \begin{array}{l} \mathcal{Y}'_i = I(X_i,t) - I(X_i + B_i\widehat{A}_k, t+1) - \widehat{\xi}_k \\ \mathcal{X}'_i = \mathcal{X}(X_i,\ I_x(X_i + B_i\widehat{A}_k, t+1),\ I_y(X_i + B_i\widehat{A}_k, t+1)) \end{array} \right. \tag{9}$$

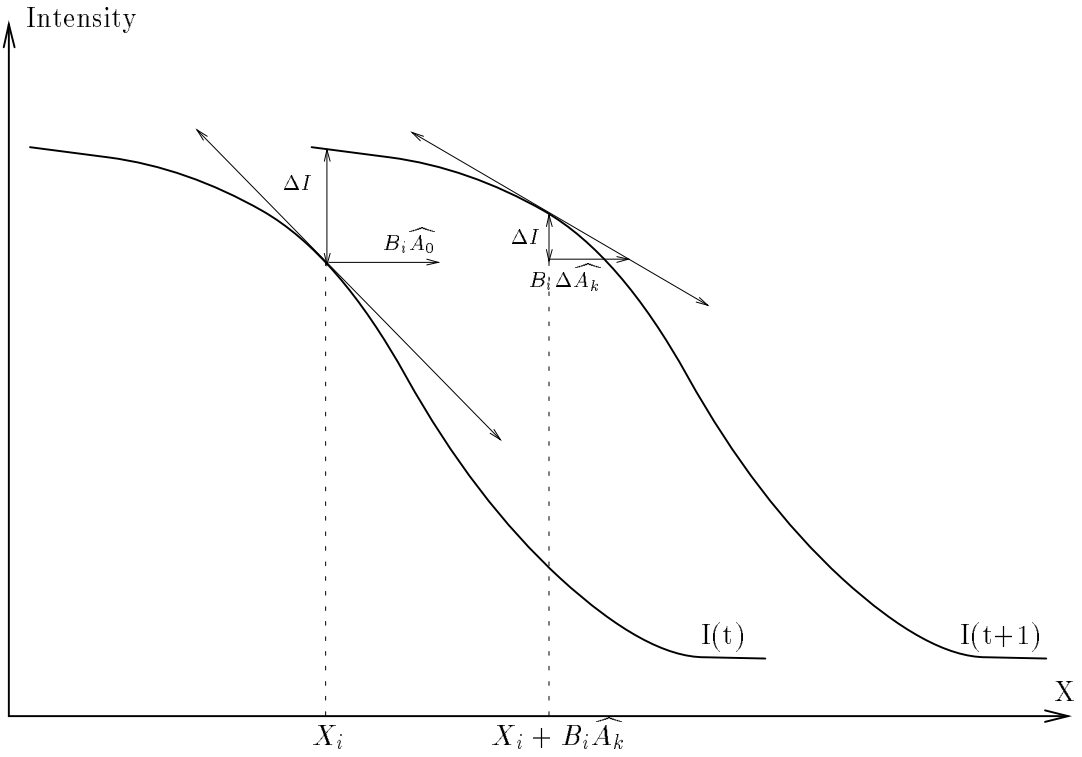


Figure 1: Geometric interpretation in one dimension of the incremental scheme

We thus minimize the following quadratic error measure :

$$E_2 = \sum_{X_i \in F} (r_i{}'')^2 \tag{10}$$

with respect to $\Delta\Theta_k$, and obtain the well-known solution:

$$\widehat{\Delta\Theta}_k = \left[\sum_{X_i \in F} \mathcal{X}_i'^t \mathcal{X}_i' \right]^{-1} \sum_{X_i \in F} \mathcal{X}_i'^t \mathcal{Y}_i' \tag{11}$$

In (9), values of I, I_x and I_y for points which are not on the 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 use an L-level Gaussian low-pass pyramid of each image [Bur84]. 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, no estimation is available and we therefore minimize the error measure:

$$E_1(\Theta) = \sum_{X_i \in F} r_i^2 = \sum_{X_i \in F} (\mathcal{X}_i \Theta - \mathcal{Y}_i)^2 \tag{12}$$

where the different terms are given by (5). At this level, displacements are small, and hence the motion constraint equation is valid. So, a first estimate of Θ^L is obtained and successive refinements using (11) are performed at the same level until the incremental estimate $\widehat{\Delta\Theta^L}$ is too small or by limiting the number of iterations. Then, the estimated parameter $\widehat{\Theta^L}$ is transmitted to the finer level, where the refinement process starts again. This is repeated until the finest level is reached. The final estimate of Θ, $\widehat{\Theta_{est}}$, is therefore the value of $\widehat{\Theta^0}$ after the last iteration at level 0. Hence, the algorithm is:

Multiresolution least mean square algorithm (MRLS) :

$\widehat{\Theta^L} \longleftarrow$ Least mean squares (residual(5))
FOR l = level L 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
$\widehat{\Theta_{est}} \longleftarrow \widehat{\Theta^0}$

Different expressions of the norm ($\|\widehat{\Delta\Theta}\|$) could be considered. The mean of the quantities $\|\vec{V}_{\widehat{\Delta A}}(X_i)\|$ over the support area F, where $\|\vec{V}_{\Delta\hat{A}}(X_i)\|$ is the incremental flow field induced by the estimated variations $\widehat{\Delta a_j}$ of the

a_j parameters. However, this is computationaly expensive, and, instead, we prefer to use a linear combination of the $\widehat{\Delta a_j}$:

$$N = \sum_j s_j \times \|\widehat{\Delta a_j}\| \tag{13}$$

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 :

$$\frac{1}{T_F}\int_F \|\vec{V}_{\widehat{\Delta a_j}}\| = s_j \times \widehat{\Delta a_j}$$

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

$$s_1 = s_4 = 1,\ s_2 = s_5 = \frac{1}{T_F}\int_F |x_i - x_F|,\ s_3 = s_6 = \frac{1}{T_F}\int_F |y_i - y_F|$$

where (x_F, y_F) is the gravity center of the region F. Thus, the coefficient s_j can be directly related to the size of the support region. Although N cannot account for the relationships between parameters in the flow field model, it is sufficient to take into account the fact that, in particular, a variation of a linear coefficient has an impact on the estimated model flow field which depends on the region size. Therefore, the criterion used in the above algorithm ($\|\widehat{\Delta\Theta^l}\| > \frac{d}{2^l}$) tests if the estimated values of the incremental parameters bring a significant modification of the 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 at the different scales.

Of course, elements in (11) are considered at current level l, I becomes I^l, X_i becomes X_i^l, support region is F^l, etc... The projection operator P performs the transformation of the motion parameters from a given level to the next finer level. It corresponds therefore to a scale change. It is easy to see that constant terms are homogeneous to a displacement, linear terms have no unit, and quadratic terms are homogeneous to the inverse of a distance. Since passing from a given level to a finer one is equivalent to multiplying the displacement by two, we can therefore infer the projection P :

$$P\ :\ \begin{cases} a^{l-1}_{const} \longleftarrow 2a^l_{const} \\ a^{l-1}_{lin} \longleftarrow a^l_{lin} \\ a^{l-1}_{quad} \longleftarrow \frac{1}{2}a^l_{quad} \end{cases} \tag{14}$$

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 time complexity. We now briefly present the two mainly used classes of robust estimators, [Hub81, Rou84].

- M-estimator: It consists in the minimization of an error sum of residuals :

$$\widehat{\Theta} = \underset{\Theta}{\operatorname{argmin}} \sum_i \rho(y_i - M(\Theta, X_i), \sigma) \tag{15}$$

  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 ([HRRS86]) 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 therefore $\psi(x) = 2x$. In this case, the influence of outliers increases linearly 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 at 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, whose ψ function is given by :

$$\psi(x, C) = \begin{cases} x(C^2 - x^2)^2 & \text{if } |x| < C\,, \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

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 number of parameters of the model.

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

$$\widehat{\Theta} = \underset{\Theta}{\text{argmin}}\ \text{Med}_i\,(\,y_i - M(\Theta, X_i)\,)^2 \tag{17}$$

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

  - if the model is nonlinear with respect to the parameters, it is impossible to estimate these parameters ;
  - if the model is linear, the computation cost is very high, increasing rapidly with the amount of data, even if a Monte-Carlo-like speed-up technique is used, [MMR91] ;
  - 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 ;

In our case, data in the support region F are more or less numerous, data noise is important (image acquisition noise, interpolation step, computation of intensity derivatives, ...) ; in addition, the model, whichever we choose (linear or quadratic), is of course 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 [HW77]. It converts this M-estimation problem into an equivalent weighted least-squares problem :

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

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 :

$$\sum_i \psi(r_i)\, \frac{\partial r_i}{\partial \Theta_j} = \sum_i w_i\, r_i\, \frac{\partial r_i}{\partial \Theta_j} = 0, \quad j = 1, \ldots, J \tag{18}$$

The weights w_i at each point X_i are therefore given by:

$$w_i = \frac{\psi(r_i)}{r_i}. \tag{19}$$

The first step of IRLS consists in obtaining a first estimate of Θ. Since the model M is linear with respect to the parameter Θ, this can be achieved using least squares minimization. Then, the weights are evaluated, and a new estimate of Θ is computed using weighted least squares. The weights are then updated and the process is repeated until convergence. In our case, this method will be embedded in a multiresolution scheme that we now describe.

Recall that the error measure we have considered in Section 3 was $\sum r_i^2$, with r_i given by (6) . It is simply reformulated here as :

$$E_3(\Theta) = \sum_{X_i \in F} \rho(r_i) \tag{20}$$

where ρ is Tukey's biweight. A scheme similar to MRLS one is used, but each minimization is now achieved using IRLS. At the coarsest level, the first IRLS (where residual of equation (5) is used) is performed as described above. However, for each of the following incremental steps (at one given level or from a level to a finer one), the first iteration of IRLS does not start with all weights w_i equal to 1, but makes use of (and adapts if necessary)

the weights derived in the last iteration of the previous IRLS step.

Up to now, the problem of estimating the standard noise deviation appearing in (15) has not been addressed. It is generally solved in parallel with the parameter estimation, and a robust estimator of this quantity usually employed is, [MMR91] :

$$\widehat{\sigma} = 1.48 \times \text{Med}_i(|r_i - \text{Med}_j(r_j)|) \tag{21}$$

This deviation can in fact be related to the constant C appearing in the ψ function (16). For instance, in [HW77], it is recommended to use a proportionality factor of 4.7 between C and σ to ensure a better efficiency in case of Gaussian noise.

In our case, since the residual can be assimilated to a displaced frame difference, C can be viewed as an intensity variation between two points in two successive images. 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 becomes 0, we will consider for C values ranging from 5 to 20. In fact, the value of this parameter mainly depends on the suitability of the motion model to the real motion : if the model is appropriate, it can be small ; otherwise, it is better to set C to a rather large value. It also depends on the local intensity variations between neighbouring entities undergoing different motion, and on the intensity texture inside those entities. In our algorithm, the C constant is chosen very large at the first estimation step (specifically, equal to the largest temporal intensity difference, in order to have an objective function convex with respect to the data). Then, it is lowered at each incremental estimation step, until it reaches in the last minimization step either a preset value (typically 8), or a value robustly estimated with (21). Experiments carried out with these two options gave similar results, but at a higher computation cost in the second case. Moreover, these experiments have shown that the value of C is not critical. Thus, we have defined a first algorithm which can be described as follows :

Robust multiresolution algorithm (RMR)

$C \longleftarrow Max(|I_t|)$
$\widehat{\Theta^L} \longleftarrow$ IRLS (residual (5),C)
FOR l = level L to level 0 DO
    iter $\longleftarrow 0$
    DO
        $C \longleftarrow f(C)$
        $\widehat{\Delta\Theta^l} \longleftarrow$ IRLS (residual(8), 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
$\widehat{\Theta_{est}} \longleftarrow \widehat{\Theta^0}$

The λ number is usually set to a small value (equal to 4 or 5), and few iterations are needed to achieve convergence at each IRLS step.

We have established a simplified version of this algorithm, using the same scheme, and exploiting the fact that the MRLS algorithm usually gives an estimation attracted by the dominant motion (and this is especially true when a constant model is used). Of course, if there are several regions undergoing different motion, or if the constraint equation is not valid (due to occlusions, high intensity contrasts, ..., see [HB93]), the estimation will be greatly disturbed. Nevertheless, the following brightness difference :

$$DFDcomp(X_i) = I(X_i + B_i\widehat{A}, t+1) - I(X_i, t) + \widehat{\xi} \tag{22}$$

is likely to be small in the region corresponding to the dominant motion. Hence, we use this information to compute a weight at each point on the finest grid after having applied the MRLS scheme once. These weights are then propagated throughout the pyramid. Then, a new coarse-to-fine estimation is performed, but instead of using least squares at each incremental estimation step, we use weighted least squares. This version avoids the IRLS computation at each incremental estimation, but several passes through the pyramid are required. In this algorithm, C is kept constant from one multiresolution estimation to the other one. This second algorithm is summarized as follows :

Pseudo-M-estimator algorithm (PSM)

$w_i \longleftarrow 1.0, \quad k \longleftarrow 0 \quad \text{and} \quad (\widehat{\Theta_{est}})_0 \longleftarrow 0$
DO
$(\widehat{\Theta_{est}})_{k+1} \longleftarrow \text{MRLS}(w_i)$
$w_i \longleftarrow \frac{\Psi(q_i,C)}{q_i} \quad \text{and} \quad k \longleftarrow k+1$
WHILE $\|(\widehat{\Theta_{est}})_k - (\widehat{\Theta_{est}})_{k-1}\| > d'$ and k$< \lambda$
$\widehat{\Theta_{est}} \longleftarrow (\widehat{\Theta_{est}})_k$

In order to filter noise, to smooth the intensity values of warped images, and to pay more attention to the points really bringing information (i.e. points of high intensity gradient), we use the following expression for q_i :

$$q_i = \frac{\sum_{X_j \in \eta_i} \|\nabla I(X_j)\| \, DFDcomp(X_j)}{\sum_{X_j \in \eta_i} \|\nabla I(X_j)\|}$$

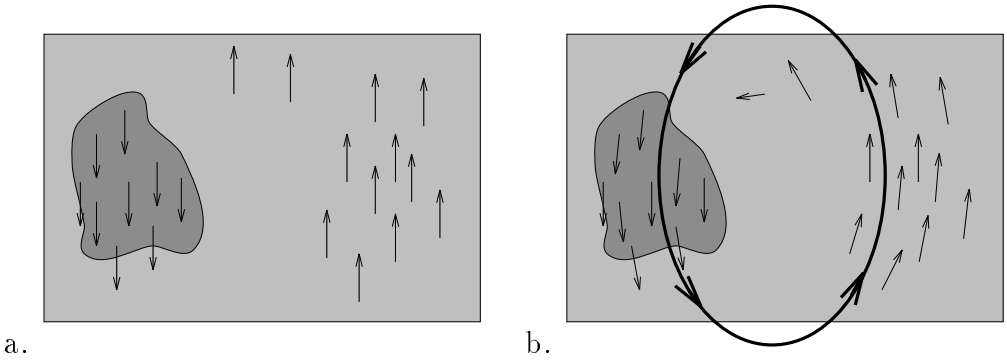
where η_i represents an 8-neighbourhood of X_i.

4.3 Complementary stages

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

1. the real motion, that we would like to recover, could be described by less parameters than those contained in 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 steps of our algorithms are based on least-squares ; therefore, the secondary objects and their motion could be "absorbed" in the minimization to sufficiently constrain the "degrees of freedom" left indeterminate by the observations corresponding to the dominant motion. This is further facilitated when the dominant motion areas are poorly textured.



a. b.

Figure 2: 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 is a rotational field.

3. since our algorithms are iterative, they greatly 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 2. Figure 2.a displays the true flow vectors, and Figure 2.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 flow 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 two estimation algorithms described in the previous section. We start the estimation process by considering a constant motion model, and we introduce a more complex model (e.g., an affine one) only after a few iterations :

Modified robust multiresolution algorithm (RMRmod)

Number of motion parameters $\longleftarrow$ 2
$C \longleftarrow Max(|I_t|)$
$\widehat{\Theta^L} \longleftarrow$ IRLS (residual (5),C)
FOR l = level L to level 0 DO
    if (l < Lc) Number of motion parameters $\longleftarrow n$
    iter $\longleftarrow$ 0
    DO
        $C \longleftarrow f(C)$
        $\widehat{\Delta\Theta}^l \longleftarrow$ IRLS (residual(8), 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
$\widehat{\Theta_{est}} \longleftarrow \widehat{\Theta^0}$

Usually, Lc is taken equal to L. This means that we start to estimate a constant motion model at the lowest resolution level, and, at subsequent levels, we consider the model chosen for the application at hand, e.g., the affine motion model. The main advantage of this method is that the incremental estimation algorithm is very efficient when a constant model is used, as explained in [BHK91]. Then, it helps to discriminate between several objects or between the background and objects. The PSM algorithm also can be modified in that way, by estimating a constant model at the very beginning with the MRLS algorithm. This leads to :

Modified pseudo-M-estimator algorithm (PSMmod)

Number of motion parameters $\longleftarrow 2$
$(\widehat{\Theta_{est}})_1 \longleftarrow$ MRLS
$w_i \longleftarrow \frac{\Psi(q_i, 2\times C)}{q_i}$
Number of motion parameters $\longleftarrow n$
DO
$(\widehat{\Theta_{est}})_{k+1} \longleftarrow \text{MRLS}(w_i)$
$w_i \longleftarrow \frac{\Psi(q_i, C)}{q_i}$ and $k \longleftarrow k+1$
WHILE $\|(\widehat{\Theta_{est}})_k - (\widehat{\Theta_{est}})_{k-1}\| > d'$ and k$< \lambda$
$\widehat{\Theta_{est}} \longleftarrow (\widehat{\Theta_{est}})_k$

The computation of the weights after the estimation of the constant motion model is performed with a constant in the ψ function twice as large as the one used in the subsequent steps. This is to take into account the fact that the model we have used is not the right one. Thus, we will keep enough points to make a good estimation of all the parameters in the multiresolution estimation. Let us note that a point with a null weight at a given iteration is not definitively discarded, and can get a non zero weight in the subsequent iterations. Finally, this method does 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 both synthetic and real motions. In the former case, we can quantitatively evaluate the results to assess the performance of the algorithms.

5.1 Experiments with real images and synthetic motion

The first experiment is the following. From the real image of Fig. 6.a and using the synthetic motion shown in Fig. 3.a, we have constructed 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). The synthetic

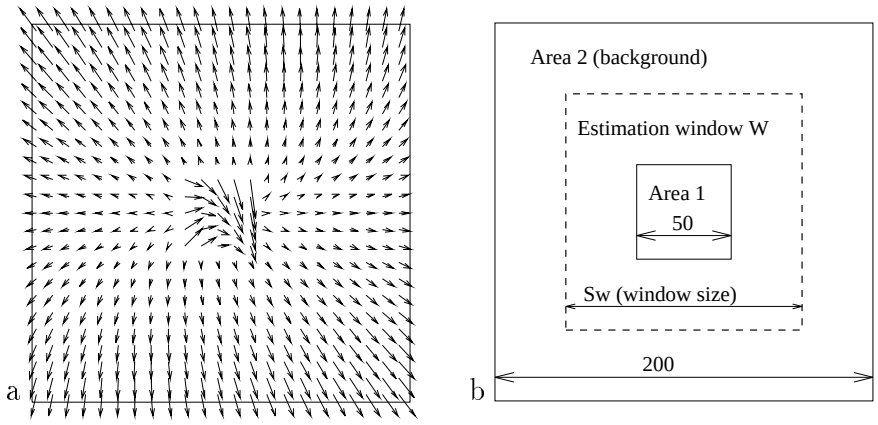


Figure 3: a) synthetic motion field; b) experimental framework ;

motion field $\vec{V}_s$ is composed of two different affine models (whose parameters are given in Table 1): the first one is applied to a square window (first area, A_1) centered in the image, and the second one to the rest of the image (second area, A_2), as illustrated in Fig. 3b.

Parameters	a_1	a_2	a_3	a_4	a_5	a_6
Model of area 1	1.0	-0.03	0.0	1.0	0.08	-0.06
Model of area 2	0.0	0.01	0.005	0.0	0.0	0.02

Table 1: Values of the synthetic motion parameters.

Then, we estimate the six parameters of the affine motion model in a square window (or support region) W of varying size S_W (see Fig. 3b), and we study the accuracy of those estimates (denoted $\widehat{A_W}$) according to the ratio of areas A_1 and A_2 within W. We consider as an adequacy measurement the average error given by ($n = 1, 2$):

$$err_n = \frac{1}{card(W \cap A_n)} \sum_{X_i \in (W \cap A_n)} \|\vec{V}_{\widehat{A_W}}(X_i) - \vec{V}_s(X_i)\|$$

Taking into account other error quantities (such as the angular error) or directly the disparity between the theoretical and estimated values of the affine parameters gives similar results and leads to the same conclusions. In these experiments, we have considered the non modified RMR and PSM

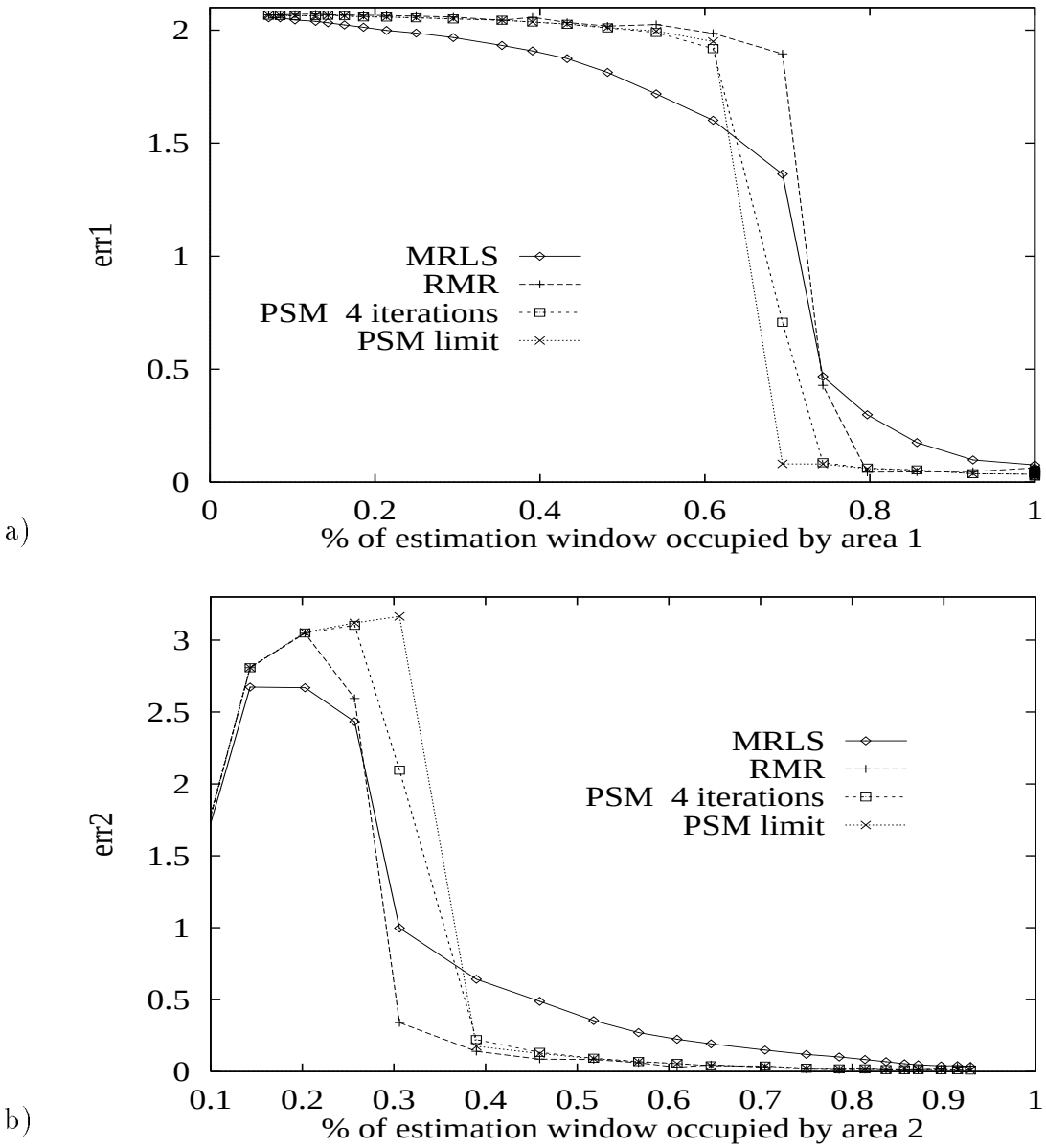


Figure 4: Average error err_n as a function of the percentage of window W occupied by : a) area 1; b) area 2. (case of the moving square in the center of the image)

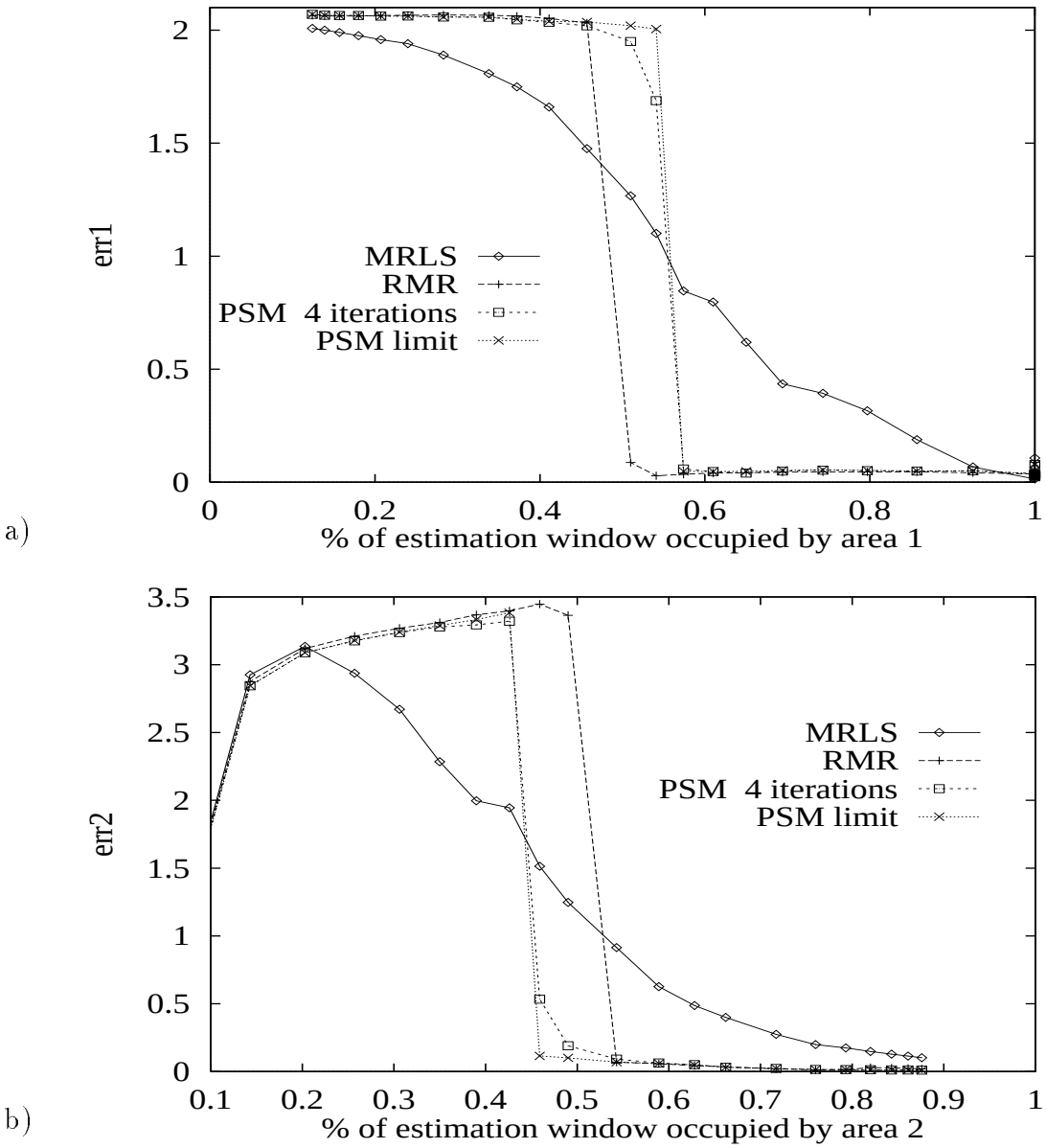


Figure 5: Average error err_n as a function of the percentage of window W occupied by : a) area 1; b) area 2. (case of the moving square situated on the car, in the foreground)

algorithms (with a value of C respectively equal to 5 and 9). Concerning the PSM algorithm, we report results obtained for a number of multiresolution estimation steps limited to four ($\lambda = 4$), and results where λ is set to infinity (called PSM limit).

Results are plotted in Fig. 4. As expected, the MRLS estimator averages the two motions (and this is even more obvious in Fig. 5), whereas both robust estimators behave quite well, recovering the dominant motion and yielding a good estimation, while the support region W is occupied by a substantial percentage of points undergoing the other motion.

However, for those two estimators, the motion of area 1 is not correctly estimated as soon as area 1 occupies less than 70% of the estimation window. In contrast, the motion of area 2 is already well estimated when area 2 only represents 40% of the support region. The transition point, separating the window sizes for which the estimated motion is that of area 1 and those for which the estimated motion is that of area 2, lies near $S_W = 62$, while area 1 represents about 65% of the support region. We would have expected a result closer to 50%. This can be explained however by the fact that the estimation is based on the motion constraint equations (4) or (8), where intensity gradient plays a crucial role : uniform areas (which occupies a non-negligible area in this example !) provide in fact no motion information. Therefore the transition point can be more adequately defined as the size ratio corresponding to :

$$\int_{W \cap A_1} \|\vec{\nabla} I\| = \int_{W \cap A_2} \|\vec{\nabla} I\| \tag{23}$$

In the first example, area 1 is located on a region of leaves of low contrast in comparison with the foreground objects (cars, pannel). The transition point corresponding to equilibrium (23) and computed at level 0 is reached for $S_W = 63$, with 63% of W occupied by area 1. To corroborate this remark, we have performed another experiment, positioning the square region A_1 on a part of the car in the middle of the lower part of the image. The same motion models as in the first experiment are used. Results are given in Fig. 5. Equilibrium (23) is now obtained when 48% of W is occupied by the first area A_1, in good agreement with plots of Fig. 5.

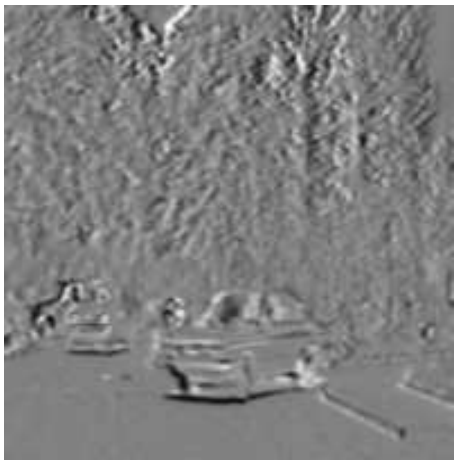
a  b 

Figure 6: "cars sequence": a) first image ; b) temporal difference between the two considered images ;

5.2 Experiments with real sequences

"Cars" sequence.

In numerous dynamic scene analysis issues, it is useful and often even necessary to first recover the camera motion, and then, to perform detection and tracking of moving objects in the scene. Figure 6.a presents the first image of the sequence. Figure (fig.6.b) shows the temporal 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 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 DFD_{comp} given by (22) is then computed using the estimated 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). Derivatives of the intensity are computed as explained in [VF92].

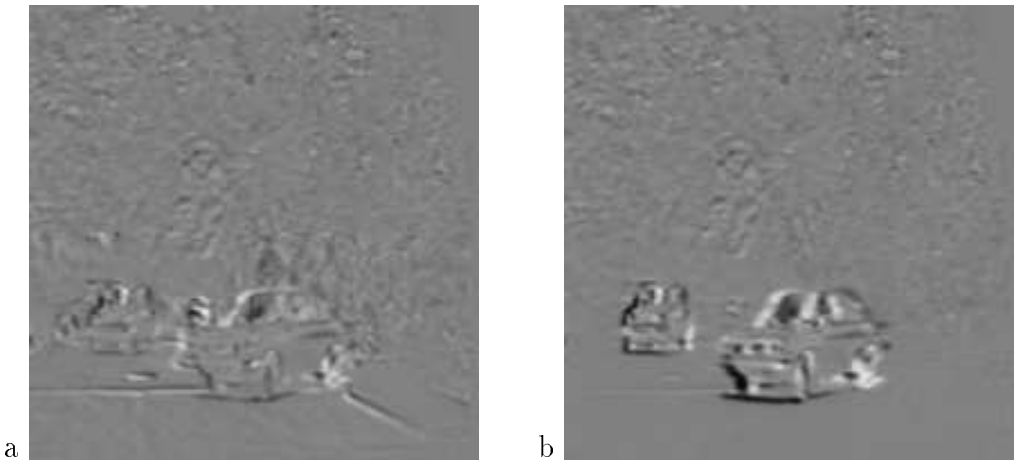


Figure 7: Compensated images DFD_{comp}, case of a velocity field estimated with : a) MRLS ; b) modified PSM ($C = 9$) ;

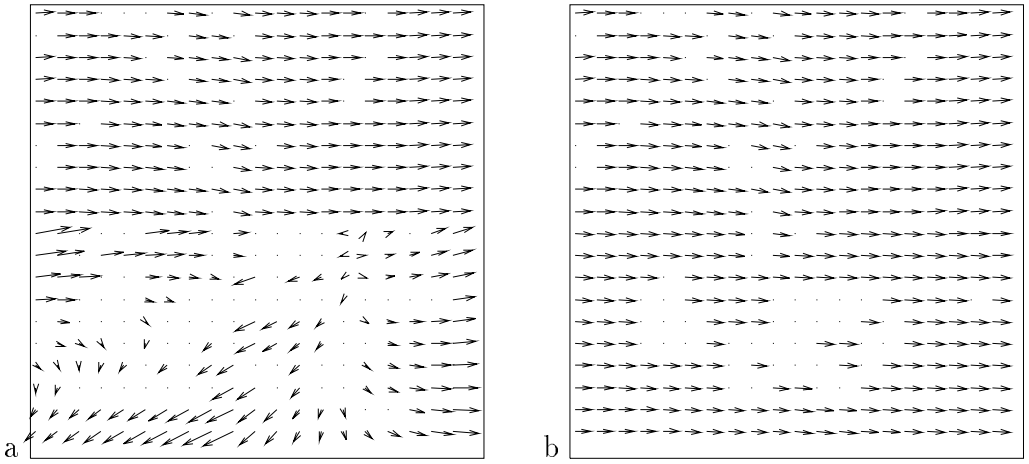


Figure 8: Velocity field corresponding to the model estimated with : a) MRLS ; b) modified PSM ($C = 9$) (For display convenience, velocity fields are subsampled by a factor of 9).

Figures 7 and 8 show results obtained with the MRLS and modified PSM algorithms. The results obtained with the modified RMR algorithm are identical to those of the PSM.

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 8 (displacement vectors are shown only at points where the dominant motion is considered as appropriate according to the value of w_i). However, in the lower blocks, the rigid motions of the two cars are coherent. Figure 7.a as well as the estimated velocity field (Fig.8.a) clearly show the averaging effect of the MRLS algorithm : the DFD is neither correct in the static part of the scene nor on the cars; the motions are also badly estimated. On the contrary, the modified PSM algorithm perfectly computes the panning motion as shown by Fig. 7.b and Fig.8.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.

Meteo sequence.

Figures 9.a and 9.b contain two successive infrared meteorological images of an equatorial area taken half-an hour apart. The scene is mainly composed of deep convective clouds giving rise to a clockwise vortex motion, whose center is located in the heart of the convective clouds (upper right corner of the central block). However, some mid-level clouds, at the right of the image are globally rotating counterclockwise, and a second convective growth in the left lower part makes some clouds go towards the bottom of the image. Moreover, there are transparent phenomena between cloud layers; the increase of the atmospheric temperature clears away parts of low- and mid- level clouds; and the deep convection results in ascending clouds leading to strong grey level variations.

The image is subdivided in nine blocks of 75×75 pixels, in which an affine motion model can be considered as a correct approximation. The estimated flow fields $\vec{V}_{\hat{A}}$ are presented whithout any post-processing step to eliminate block effects. The MRLS algorithm obtained globally correct results

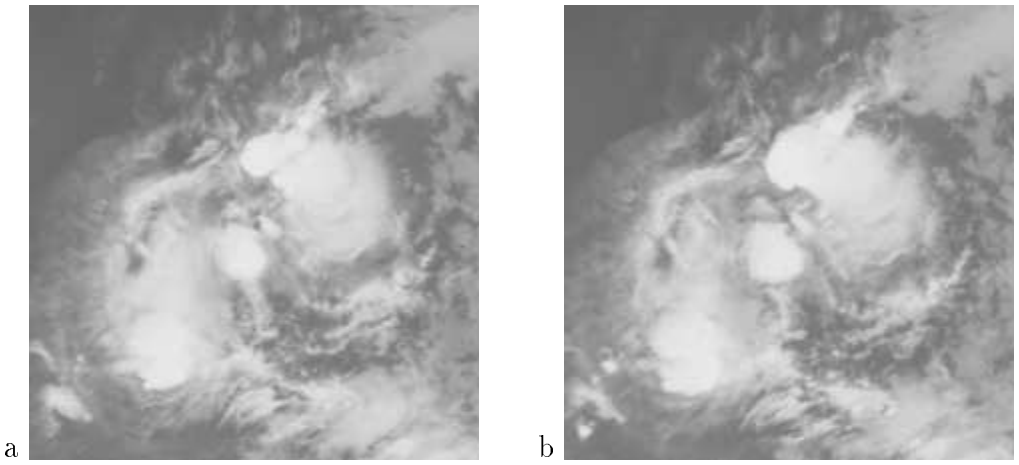


Figure 9: Meteo Sequence : a) first and b) second image

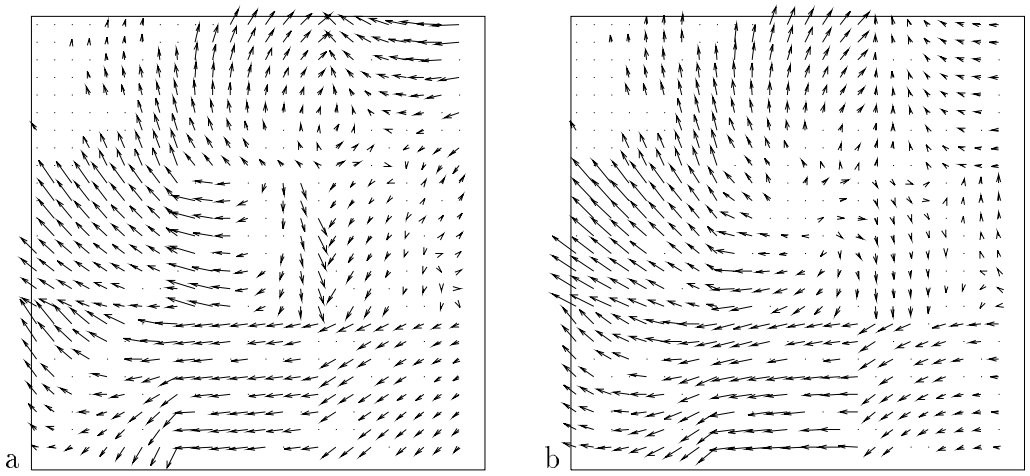


Figure 10: Velocity field computed with : a) MRLS; b) modified PSM ($C = 18$);

(Fig. 10.a), but still suffers from several limitations, in particular the motion is badly estimated in the central block. On the other hand, the modified PSM algorithm, although it computes a translation at the first multiresolution estimation, recovers the rotational motion quite well in that block, and block effects are significantly reduced. Values of parameter ξ for each block can be found in Table 2; this shows it is worth introducing it in the model. A value of $\xi > 0$ indicates an average decrease of the grey levels in the block. Almost

all estimated values are positive, which is coherent with the clearing of some clouds throughout the sequence.

0.5	2.0	-1.1
3.4	2.3	2.6
0.7	-0.6	4.0

Table 2: ξ values estimated in each block

As for the "cars" sequence, the modified RMR algorithm gives very similar results. However, its computational cost is usually slightly less important than the one of the modified PSM algorithm.

Conclusion

We have described in this report two robust multiresolution parametric motion model estimation algorithms. They were compared favorably 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 using an explicit segmentation step. They can also be seen as an efficient first step of a motion segmentation scheme or of a motion field estimation scheme.

This study is supported in part by the French Ministry of Research in the context of the GDR-PRC "Man-Machine Interface"(MRT contract 91S269), and by "Région Bretagne"(Brittany Council) through a contribution to student grant. The authors would like to thank D. Dagorne from Centre de Météorologie Spatiale, Lannion, for providing the meteorological image sequence.

References

[AN88] 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.

[BAHH92] J.R. Bergen, P. Anandan, K. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In *Proc. of ECCV-92, S.Margherita Ligure, Italy*, pp 237–252, Springer-Verlag, 1992.

[BAK91] 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.

[BBHP90] 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, Osaka*, pp 27–32, Dec. 1990.

[BD93] B. Bascle and N. Deriche. Stereo matching, reconstruction and refinement of 3D curves using deformable contours. In *Proc. 4th Int. Conf. Computer Vision, Berlin*, pp 421–430, May 1993.

[BF93] 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.

[BHK91] 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, Princeton*, pp 187–193, October 1991.

[Bla92] M. J. Black. *Robust incremental optical flow*. PhD thesis, Yale University, Computer Science Dept, September 1992.

[Bur84] P.J. Burt. The pyramid as a structure for efficient computation. In A. Rosenfeld, editor, *Multiresolution Image Processing and Analysis*, pp 6–35, Springer-Verlag, 1984.

[DP91] T. Darell and A. Pentland. Robust estimation of a multi-layered motion representation. In *Proc. IEEE Workshop on Visual Motion, Princeton*, pp 173–178, Oct. 1991.

[Enk88] 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.

[HB93] 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, No.12, Dec. 1993.

[Hoe89] M. Hoetter. Differential estimation of the global motion parameters zoom and pan. *Signal Processing*, Vol.16:249–265, 1989.

[HRRS86] 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.

[HS81] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, Vol.17:185–203, 1981.

[Hub81] P.J. Hubert. *Robust statistics.* Wiley, 1981.

[HW77] P.W. Holland and R.E. Welsch. Robust regression using iteratively reweighted least squares. *Commun. Stat.- Theor. Meth.*, A6:813–828, 1977.

[JMB91] J.M. Jolion, P. Meer, and S. Batauche. Robust clustering with application in computer vision. *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol.13:791–802, August 1991.

[KD88] J. Konrad and E. Dubois. Multigrid Bayesian estimation of image motion fields using stochastic relaxation. In *Proc. 2nd Int. Conf. Computer Vision, Tarpon Springs, Florida*, pp 354–362, Dec. 1988.

[MB92a] F. Meyer and P. Bouthemy. Estimation of time-to-collision maps from first order motion models and normal flows. In *Proc. 11th Intern. Conf. on Pattern Recognition, The Hague*, pp 78–82, 1992.

[MB92b] F. Meyer and P. Bouthemy. Region-based tracking in an image sequence. In G. Sandini, editor, *Proc. of 2nd ECCV-92*, pp 476–484, Springer-Verlag, 1992.

[MB92c] A. Mitiche and P. Bouthemy. *Computation and analysis of visual motion : a review.* Technical Report No 92-26, INRS-Télécommunications, Montréal, October 1992.

[MMR91] 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.

[Nag88] H.H. Nagel. From image sequences towards conceptual descriptions. *Image and Vision Computing Jal*, Vol.6, No.2:59–74, May 1988.

[NL91a] 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*, pp 132–139, Oct. 1991.

[NL91b] H. Nicolas and C. Labit. Global motion identification for image sequence analysis and coding. *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, Toronto*, Vol. IV:2825–2828, May 1991.

[Rou84] P.J. Rousseeuw. Least median of squares regression. *Jal of the American Statistical Association*, Vol.79, No 388:871–880, Dec. 1984.

[SM89] J. Schmetz and M.S. Mhita. Diurnal and interdiurnal variability of IR and WV brightness temperatures from Meteosat. *ESA Journal*, 13:329–341, 1989.

[VF92] T. Vieville and O. Faugeras. Robust and fast computation of unbiased intensity derivatives in images. In *Proc. 2nd European Conf. on Computer Vision, Genova*, pp 203–212, May 1992.

[WK93] 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.

[ZQY89] 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.