Efficient Half–Quadratic Regularization with Granularity Control

Mariano Rivera and Jose L. Marroquin
Centro de Investigacion en Matematicas A.C.
Apdo. Postal 402, Guanajuato, Gto. Mexico 36020
mrivera@cimat.mx

September 11, 2001

Abstract

In the last decade, several authors have proposed edge preserving regularization (EPR) methods for solving ill posed problems in early vision. These techniques are based on potentials derived from robust M–Estimators. They are capable of detecting outliers in the data, finding the significant borders of a noisy image and performing an edge–preserving restoration. These methods, however, have some problems: they are computationally expensive, and often produce solutions which are either too smooth or too granular (with borders around small regions). In this paper we present a new class of potentials that permits separate control of robustness and granularity, producing better results than the classical ones in both scalar and vector-valued images. We also present a new fast, memory-limited minimization algorithm.

Contents

1 INTRODUCTION

In the fields of image processing, image analysis and computer vision, one deals with the problem of reconstructing an image $\hat{f}$ from noisy and degraded observations g. Consider the following model of the observations:

$$g = F\left(\hat{f}\right) + \eta; \tag{1}$$

where η is additive noise and F is a linear operator that is assumed to be known. For example in optical blurring, F corresponds to the convolution with the Point Spread Function (PSF) of the imaging system. In other cases, F can be a linear approximation of a non-linear transformation (e.g. in the computation of optical flow).

The information provided by the data and the direct model (1) is (in general) not enough for an accurate estimation of $\hat{f}$, so that the regularization of the problem is necessary. That means that, *a priori* information or assumptions about the structure of $\hat{f}$ need to be introduced in the reconstruction process. The regularized solution f^* is computed by minimizing an energy functional U:

$$f^* = \arg\min_f U(f) \tag{2}$$

where

$$U(f) = D(f) + \lambda R(f). \tag{3}$$

The data term D establishes that the reconstruction f should be consistent with the data g: The regularization term R imposes a penalty for violating the *a priori* assumptions, and the relative contribution of each term to the global energy is weighted by the parameter λ. The Data term can been written as

$$D(f) = \sum_r \rho_D\left(t_r(f)\right)$$

where t is the residual error defined by

$$t_r(f) = F(f)_r - g_r;$$

and $r = (x; y)$ represents a site in the pixel lattice L, and ρ_D is a potential function that defines the residual norm; the subindex denotes that this potential is associated with the data term. In the framework of Bayesian regularization [1], D is chosen as the negative log–likelihood and the prior constraints are incorporated in the form of *a priori* (Markov Random Filed) model for f, so that $R(f)$ takes the form of a sum, over the cliques of a given neighborhood system, of a set of "potential functions" supported on those cliques. One may take for instance as the neighborhood N of a pixel r its 8 closest neighbors:

$$N_r = \{s : |r - s| < 2\}$$

and cliques of size 2 $\langle r,s\rangle$ that correspond to horizontal, vertical and diagonal pixel pairs, so that $R(f)$ takes the form:

$$R(f) = \sum_{r} \sum_{s \in N_r} \rho_R(u_{rs}(f)).$$

where the residual error for the regularization term is defined by

$$u_{rs}(f) = f_r - f_s. \qquad (4)$$

and ρ_R is a potential function. For instance, the homogeneous spring model R_S is obtained by assuming that η corresponds to Gaussian noise:

$$\rho_D(t_r(f)) = (F(f)_r - g_r)^2. \qquad (5)$$

and choosing ρ_R as a quadratic potential over the first differences:

$$\rho_R(u_{rs}(f)) = d_{rs}(f_r - f_s)^2 \qquad (6)$$

with

$$d_{rs} = |r - s|^{-1}.$$

This quadratic potential corresponds to the *a priori* assumption that the original data $\hat{f}$ is globally smooth. Then, assuming that F is linear, the cost functional that results from potentials (5) and (6) is quadratic:

$$U_1(f) = \sum_{r} \left\{ |t_r(f)|^2 + \lambda \sum_{s \in N_r} d_{rs}|u_{rs}(f)|^2 \right\} \qquad (7)$$

This cost functional is not robust to outliers. In the image restoration context, the outliers are located at those sites where the assumptions implicit in the cost function are not fulfilled. In particular, for a regularization term that assumes global smoothness, the outliers correspond to the edges in the image. As a consequence, the potential function (6) will produce an over–smoothing of the real edges of the image.

To alleviate this problem, there have been proposed robustpotential functions for the data and regularization terms. This regularization technique is usually based on potentials derived from robust M–Estimators and is capable of detecting outliers in the data, finding the significant edges of a noisy image and performing an edge–preserving restoration. However, there are some problems: the robust potentials that are in use have a single parameter that controls the minimum residual magnitude that corresponds to an outlier; it often happens in noisy images that if this parameter is too small, many small regions generate edges around them so that the solutions appear granular, while if the value of this parameter is increased, some true edges are not preserved Another problem is that the convergence of the algorithms that have been proposed for the minimization of the corresponding cost function are relatively slow, making these methods computationally expensive.

The propose of this paper is to present a formulation for robust potentials that produces faster and better behaved algorithms that result in better reconstructions and a significant time processing reduction.

The organization of the paper is as follows: in section II, a brief introduction to robust regularization is presented. Section III introduces the new formulation for robust potentials. Also, in that section, we show how to write robust energy terms for vector valued data. In section IV, we present a non-linear Conjugated Gradient algorithm for half–quadratic regularized functionals with minimal memory requirements. Experiments that demostrate the performance of the new potentials and the algorithms introduced herein are presented in section V. Finally, our conclusions are given in section VI.

2 REVIEW OF ROBUST REGULARIZATION BASED ON M-ESTIMATORS

To solve the problem of over–smoothing the real borders in f, potential functions for the data and regularization terms that increase at a smaller rate than the quadratic potential have been used [3][4][5][6][7][8][11][10]. These robust potentials $\rho(t)$ (that correspond to robust objective functions used in the statistical literature[7],) have the following characteristics [8][12][13]:

1. $\rho(t) \geq 0 \ \forall t$ with $\rho(0) = 0$;
2. $\rho'(t) \equiv \partial\rho(t(f))/\partial f$ exists.
3. $\rho(t) = \rho(-t)$, the potential is symmetric
4. $\frac{\rho'(t)}{2t}$ exists
5. $\lim_{t \to +\infty} \frac{\rho'(t)}{2t} = 0$
6. $\lim_{t \to 0^+} \frac{\rho'(t)}{2t} = M,\ 0 < M < +\infty.$

(8)

Condition (8.1) establishes that a residual equal to zero must produce the minimum cost. Condition (8.2) constrains ρ to be differentiable, so that one can use efficient deterministic algorithms for minimizing the cost function (really, to compute a local minimum). Condition (8.3) constrains ρ to penalize equally positive and negative values of t. Finally, conditions (8.4 to 8.6) imposes the robustness condition. A robust potential corresponds to (in general) a non-convex potential that grows at a slower rate than the quadratic one. A local minimum of the robust potential based energy functional

$$U_2(t) = \sum_r \left\{ \rho_D(t_r(f)) + \lambda \sum_{s \in N_r} \rho_R(u_{rs}(f)) \right\} \quad (9)$$

is computed by solving the system

$$\psi_D(t_r(f)) + \lambda \sum_{s \in N_r} \psi_R(u_{rs}(f)) = 0. \qquad (10)$$

where the derivatives

$$\psi_D(t_r(f)) = \partial\rho_D(t_r(f))/\partial f_r \quad \text{and} \quad \psi_R(u_{rs}(f)) = \partial\rho_R(u_{rs}(f))/\partial f_r$$

are called the influence functions. One can also define:

$$\omega_r = \frac{\psi_D(t_r(f))}{2t_r(f)} \quad \text{and} \quad \omega_{rs} = \frac{\psi_R(u_{rs}(f))}{2u_{rs}(f)} \qquad (11)$$

as the weight functions. Then (as was shown in [4][5][8][12]), one can solve the non–linear system (10) with the following two–step iterative algorithm:

Algorithm 1 Weighted linear system

```
Given an initial guess for $f$;
Repeat:
    1. Compute the weights $\omega_r$ and $\omega_{rs}$ using (11).
    2. Solve the system
           $$\omega_r t_r(f) + \sum_{s \in N_r} \omega_{rs} u_{rs}(f) = 0$$
       for $f$, keeping $\omega_r$ and $\omega_{rs}$ fixed.
Until convergence.
```

(12)

The ARTUR algorithm reported in [8] corresponds to algorithm 1 but using as starting point for f an homogeneous image equal to zero.

In References [7] and [8] one can find pictorial summaries of the robust potential function used in robust regularization. One may define a classification of potentials functions based on the shape of the influence functions ψ [13]:

1. Monotone ψ (MT). $\psi(t)$ is constant for $|t| \geq \theta$. Where θ is a given threshold An example is the Huber´s potential function [12]:

$$\rho_h(t) = \begin{cases} t^2 & |t| < 1/2 \\ (|t| - (1/2)^2) & \text{otherwise} \end{cases}$$

2. Soft redescender ψ (SR). One has $\psi(\infty) = 0$. An example is the Cauchy´s potential function:

$$\rho_c(t) = \log(1 + t^2)$$

3. Hard redescender ψ (HR). One has $\psi(t) = 0$ for $|t| \geq \theta$; where θ is a given threshold. This kind includes the widely used Welsch and Tukey (bi-weight) potential functions. For example the Tukey´s potential function is defined as:

$$\rho_b(t) = \begin{cases} (1 - (1 - (2t)^2)^3) & |t| < 1/2 \\ 1 & \text{otherwise:} \end{cases}$$

Plots of the four classes of potential functions are shown in Figure 1. The corresponding influence functions and the weight functions are also illustrated. In order to choose the "right" ρ–function for computer vision problems, we need take into account the following:

1. Robustness to outliers. That is, one wants to reduce the effect of large errors in the solution.

2. Computational difficulties. The minimization algorithm must be stable and have fast convergence.

In the robust statistics literature [16] one can find measures of robustness for potential functions. These measures are derived from the influence function; they show the influence of large errors (outliers) and small errors (measurement errors); other common measure is the breakdown point (that in general does not depend on the shape of the potential ρ) that measures the robustness with respect to large quantity of outliers. The HR (potentials that reject extreme outliers completely) and SR (potentials that reduce considerably the effect of extreme outliers) classes are preferred for regression and estimation problems in both contexts: statistics [13][15][16] and image processing–computer vision [3][4][6] [7][8][10][17][18][19][20][21].

For the case that concerns this paper (image restoration), the robust methods are capable of detecting the significant outliers in the data, finding edges (outliers with respect the prior assumption) of a noisy image and performing an edge–preserving restoration. However, the HR–potentials do not guarantee uniqueness of the solution [7]. As a consequence, a good initial guess must be provided in order to avoid to be trapped by a "bad" local minimum. Another problem is that there is no explicit control of the granularity of the solution, and as a result one may have either inaccurate definition of edges or an over detection of small details (see Figure 4-c). Finally, the convergence rate of the minimization algorithm 1 is relatively slow, so that it is computationally demanding.

In an attempt to solve these problems, in [8] the initial point is chosen as an homogeneous image equal to zero (algorithm ARTUR). This reduces the granularity, but it also reduces the accuracy of the computed edges (see Figure 4–f) and does not contribute to the reduction of the computation time. Another improvement, introduced in [7], is to make the potential in the data term robust, which allows one to eliminate unstructured outliers; this however, may cause Algorithm 1 to become unstable because the weights ω_r and ω_{rs} can both be equal to zero (because of condition (8.5)), or produce ill–conditioned systems for small values of these weights. Also, this formulation does not consider that in the image processing context, outliers in the data may be structured (there may be small and well defined regions, see Figure 4–c), so that it is not clear how to constrain granularity. In [7][9][10] and [18] potentials that penalize thickness and promote the continuity of the borders are introduced. The result is a better definition of the edges, but the computational time is increased because there is no closed formula for the weights, so that they must be computed for each iteration as the solution of a non–linear system which causes a significant slow–down of the algorithm.

3 POTENTIALS FOR EDGE PRESERVING REGULARIZATION WITH GRANULARITY CONTROL

In this section we propose a method for stabilizing edge-preserving potentials that produces faster and better behaved algorithms. First, we note that condition (8.5) needs to be redefined in order to avoid that the weights ω_r and ω_{rs} can both be close to zero, ill–conditioning the system (10). The new condition is:

$$\lim_{t \to +\infty} \frac{\rho'_R(t)}{2t} = \mu \qquad (13)$$

where $\mu \in (0;1]$ is a positive parameter. Now, the granularity of the solution has to be controlled, so that a large and well defined region is preferred over a group of small regions (well defined regions are such that the weights at their borders are zero if a HR–potential is used), this is illustrated in Figure 2. In order to introduce this control in the robust potential, it is necessary to include an additional term in the cost functional (9), that assigns a small additional cost to large jumps that are assigned a constant cost by the potential ρ_Q. An easy way to stabilize the system and at the same time control granularity is simply to add a quadratic term to the HR–potential ρ_H, so that it has heavier "tails". The resulting regularization potential ρ_Q is given by

$$\begin{aligned} \rho_Q(u_{rs}(f);k_2;\mu) &= \mu d_{rs}[u_{rs}(f)]^2 \\ &\quad +(1-\mu) d_{rs}\rho_H(u_{rs}(f);k_2); \qquad (14) \end{aligned}$$

where the μ parameter controls the granularity of the solution and k_2 is a positive scale parameter. Note that for the extreme value $\mu = 1$, ρ_R is quadratic (non-robust) and strongly penalizes the granularity. When μ is close to zero 0; the potential is HR–robust and promotes piecewise smooth reconstructions. Small values for μ allow smooth changes inside the regions and control the size of the grain in the reconstruction. The form of the ρ_Q potential together with its influence and weight functions is presented in Figure 3. These types of influence functions corresponds to the so–called Quasi–robust estimators in the statistics literature [22]. In all our experiments we use

$$\rho_H(x;k) = 1 - \frac{1}{2k}\exp(-kx^2);$$

where k is the scale parameter, and for the data term we set

$$\rho_D(t;k) = \rho_H(t;k);$$

then we have:

$$U_N(f) = \sum_r \left\{ \rho_H(t_r(f);k_1) + \lambda \sum_{s\in N_r} \rho_Q(u_{rs}(f);\mu;k_2) \right\}; \qquad (15)$$

where k_1 is the scale parameter for the data term. The minimization of (15) with respect to f is computed by solving the non–linear system

$$\frac{\partial}{\partial f_r}\left\{ \rho_H(t_r(f);k_1) + \lambda \sum_{s\in N_r} \rho_Q(u_{rs}(f);k_2;\mu) \right\} = 0$$

this non–linear system can be written in matrix form:

$$W_f f = \hat{g} \qquad (16)$$

where the matrix W_f depends on f and $\hat{g}_r = [\omega_D]_r g_r$; with ω_D as the corresponding vector of weights for the data term. As ρ_Q satisfies (8) but (8.5) is replaced by (13), then, (16) cannot become singular. In section IV, we present an efficient Conjugate Gradient algorithm for solving (16), which has a better convergence rate than Algorithm 1.

3.1 Robust Regularization for Vectorial Data

In the restoration or analysis of vector-valued images $\bar{f} = [f_1; f_2; \ldots; f_M]^T$ (e.g. optical flow and color images processing) given the data $\bar{g} = [g_1; g_2; \ldots; g_M]^T$; it is necessarily to couple the process over the M channels.

For illustration proposes, we assume that the channels of $\bar{f}$ are independently acquired, so that the outlier rejection in the data term is decoupled. On the other hand, the outliers corresponding to the regularization term are coupled because we expect that the joint contribution of all channels results on a better detection of the edges. Then the robust potentials for the data and the regularization terms (respectively) are

$$D(\bar{f}) = \sum_m \sum_r \rho_H(t_r(f_m);k_1)$$

and

$$R(\bar{f}) = \sum_{m}\sum_{r}\sum_{s\in N_r} \widehat{\rho}_R\left(\left|u_{rs}\left(\bar{f}\right)\right|; k_2; \mu\right)$$

with $m = 1,\ldots,M$; where $\left|u_{rs}\left(\bar{f}\right)\right|^2 = (f_{1,r} - f_{1,s})^2 + (f_{2,r} - f_{2,s})^2 + \ldots + (f_{M,r} - f_{M,s})^2$. The solution $\bar{f}$ is computed by solving

$$\frac{\partial}{\partial f_{m,r}}\left[\rho_H\left(\frac{t_r(f_m)}{k_1}\right) + \sum_{s\in N_r}\widehat{\rho}_R\left(\left|u_{rs}\left(\bar{f}\right)\right|; k_2; \mu\right)\right] = 0. \qquad (17)$$

4 MINIMIZATION ALGORITHM

The robust cost functionals presented above are non–quadratic functions with a large number of variables. The minimization algorithms based on the iterative solution of a weighted linear system [e.g., based on Algorithm 1 (10)] as the one used in [4][8] are relatively slow. This is particularly critical in the case of processing three–dimensional (3D) data as in the case of 3D image registration [21]. If one wants to accelerate the computation of the solution, the use of specialized algorithms for directly solving the non–linear equation system (16) (as the non–linear conjugate gradient (NLCG)[23] or the Newtonian type ones[24],) are not a good choice. The reason its that, for example, in order to guarantee convergence, the NLCG algorithms require that at each iteration, the partial solution guarantees a sufficient reduction in the cost $U(f)$ [25], that is

$$U(f_{n+1}) \quad (1-\varepsilon)U(f_n).$$

where $\varepsilon \in (0,1)$ is a small positive constant. In order to satisfy this constraint, the NLCG algorithm must perform the expensive evaluation of the cost function at each iteration. On the other hand, the Newtonian type algorithms (say, the Gauss–Newton method), additionally need to compute the product of an approximation to the Hessian of $U(f)$ and a vector, this computation being more expensive than evaluating the energy.

This section presents an efficient conjugate gradient algorithm for minimizing (computing a local minimum of) non–quadratic functionals $U(f)$ that are sums of half-quadratic potentials. Before to introducing the HQCG algorithm, we analyze the alternated minimization strategy used by Algorithm 1.

If the a potential ρ fulfills conditions (8) in its original form or with (13) instead (8.5), then there exists a function ψ that penalize an over detection of outliers; in this case ρ has an equivalent half–quadratic formulation:

$$\rho(z) = \min_{\omega} \tilde{\rho}(z;\omega);$$

where $\tilde{\rho}(z;\omega) = z^2\omega + \psi(\omega)$ [4][5][7][8]. In the half–quadratic formulation, one may consider that the weights [given by (11)] result from minimizing the potential $\tilde{\rho}$ w.r.t. ω. Thus, Algorithm 1 is performing by the alternated minimization w.r.t. t, ω_r, and ω_{rs}. This means that the non-linear system (16) is solved by the iterative scheme:

$$W_{f^t} f^{t+1} = \hat{g}^t. \tag{18}$$

In a half-quadratic sense, the computation of the matrix W_{f^t} corresponds the minimization w.r.t. the weights.

The problem with the alternated minimization strategy is that the computation of the weight matrix W and the minimization of the weighted linear system (18) are decoupled, which decreases the efficiency of the method.

The method we propose is based instead in the direct minimization of (15) using non–linear conjugated gradient. This general algorithm, however, can be made more efficient, in this case taking advantage of the half–quadratic structure; in particular, this structure allows one to derive a formula for the optimal step size at each iteration: since the minimizer of (15) satisfies (18), where W_{f^t} is a positive definite matrix, the optimal step size

$$\alpha_n = \min_{\alpha} U(f_n + \alpha s_n)$$

may be computed as

$$\alpha_n = \frac{r_n^T r_n}{s_n^T W_{f^t} s_n},$$

where r_n and s_n are the currents residual vector and descent direction, respectively.

The resulting HQCG algorithm is:

Algorithm 2 HQCG

```
Set $n = 1, \beta = 0, f_0$ equal to an initial guess.
Repeat:
   1. Set $A_n = W_{f_n}$.
   2. $r_n = \hat{g}_n - A_n f_n$
   3. if($n \neq 1$)   $\beta = \max\left\{0, \frac{r_n^T (r_n - r_{n-1})}{r_{n-1}^T r_{n-1}}\right\}$
   4. $s_n = r_n + \beta s_{n-1}$
   5. $\alpha_n = \frac{r_n^T r_n}{s_n^T A s_n}$
   6. $f_{n+1} = f_n + \alpha_n s_n, n = n + 1.$
      Until $|r_n| < \varepsilon$.
```

Now, we explain each step in detail. Step one corresponds to updating the weights $!_r$, and $!_{rs}$, i.e., one step of the alternated minimization in the half–quadratic sense. The gradient r is updated in step 2. As system A is changing at each iteration, we cannot guarantee that the gradient at iteration n is normal to the last descending direction $s_{n ¡ 1}$(i.e. in general $r_n^T s_{n ¡ 1} 6= 0$); so that we compute ¯ with the Fletcher–Reeves (FR) formula with restarting, which is the one generally used in the NLCG algorithms. This FR formula has demonstrated to have better performance in the case of non–linear systems than the Polak–Riviere formula [26].

Note that since we are computing the optimal step size, the condition

$$U(f_n + ®_n s_n) \quad (1 ¡ ")U(f_n):$$

is automatically satisfied.

5 EXPERIMENTS

We present a set of experiments for illustrating the performance and viability of the robust potential with granularity constraint. To isolate the effect of k_2 and ¹, for these experiments we kept the value of k_1 at $k_1 = 1:0$ (so that $½_D$ is in fact a non–robust potential)

5.1 Segmentation

In the first set of experiments, we study the influence of the 2 parameters that control edge preserving and granularity [i.e. k_2 and ¹; respectively in (14)] in the filtered image. The input image (Fig. 4–a) is an axial section of a magnetic resonance image of the brain, with the gray levels normalized in the interval [0,1]. The task in this case is to segment the brain from non–brain tissue, eliminating as much unwanted detail as possible, without distorting the position of the brain /non-brain (B–NB) boundary.

The starting point for HQCG algorithm was the original image. In all the cases we used $¸ = 200$. Fig. 4–b shows the restoration with $k_2 = 80$; and $¹ = 0$ (maximum granularity). As one can see, there is a lot of unwanted detail. If the edge preserving parameter k_2 is decreased (by setting $k_2 = 60$ and $¹ = 0$) the image of the Fig. 4–c is obtained. Note that some of detail is eliminated (although not completely), but the B–NB boundary is lost. On other hand, if ¹ is increased to 0:03 (for $k_2 = 800$) one obtains the image of Fig. 4-d, in which the unwanted detail is smoothed out without dislocating the B–NB boundary. The evaluation of the weight function for the potential $½_Q(u_{rs}(f); ¹; k_2)$ is shown in Fig. 4–e. As a comparison, in Fig. 4–f we present the filtered image with the ARTUR algorithm [8], using the Geman–McClaude potential $½_{GM}(t) = t^2=(k_2+t^2)$ [3] in the regularization term, which is considered the one that gives the best results [8]. The parameters where hand–adjusted to get the best possible results; their value was $¸ = 10; k_2 = 600$. In this case the starting point was an homogeneous image equal to zero, as is recommended in [8].

5.2 Denoising Vectorial Images

We illustrate the effect of the operation over vectorial images by denoising a color image (channels red (r), green (g) and blue (b)). First row in figure 7 shows the RGB channels of a color image. Second row shows the result of processing every channel independently with an HR–potential in the regularization term (¸ = 50, k_2 = 300, and ¹ = 0). As one can appreciate, the edges are not well detected and the images are still noisy. Third row shows the effect of coupling the HR–potential over all the channels (¸ = 50, k_2 = 75, and ¹ = 0). In spite the fact that there is an improvement in the computed images, they are still noisy and the borders are not well preserved. Finally, fourth row shows the computed images that result from coupling the processing over the three channels and considering a penalization over the granularity (¸ = 50, k_2 = 300, and ¹ = 0:01). The improvement is evident: the edges are well preserved and the noise is removed.

5.3 Performance of the HQCG Algorithm

Here we present a comparison of the performance of the algorithms HQCG and ARTUR. The conditions of the test correspond to the recommended in Ref. [8] for the ARTUR algorithm: the starting point was an homogeneous image equal to zero, which contributes to give non–granular results [8], so granularity was not constrained in this experiment; at each iteration both algorithms "introduce" edges. The parameters were ¸ = 30, k_2 = 600 and ¹ = 0. The minimization of the half-quadratic cost in ARTUR algorithm was computed by performing 20 iterations of the linear conjugate gradient algorithm (ARTUR–CG).

In figure 6, panel (a) shows the real image used for the test. Panels 6–b, 6–c shown the corresponding filtered images after 30, and 70 iterations (respectively) of the HQCG algorithm. Panel 6–d the resulting image after 300 iteration of the ARTUR algorithm. One can note that the dial disc of the telephone is starting to be defined in panel (a) while in the iteration 300 of ARTUR algorithm [panel (d)] is still imperceptible; besides, the edges in the soccer ball are better defined and allocated by HQCG than by ARTUR. The computation times in Fig. 6 correspond a pentium III at 800 Mhz based workstation. The evaluation of the cost function versus the iteration number is plotted in figure 7. As one can see, the HQCG presents a fast convergence rate and a smoother transition between iterations. Note however that one HQCG iteration takes 1.5 times an ARTUR iteration.

5.4 Computation of the Optical Flow

The last experiment illustrates the performance of the presented technique in a classical computer vision problem: the computation of optical flow (OF).

Introducing in the Horn-Shunck [27] formulation for OF the robust potential, one gets the energy functional:

$$U(\overline{w}) = \sum_{r \in L} \left\{ \rho_H \left(t_r(\overline{w}); k_1\right) + \lambda \sum_{s \in N_r} \hat{\rho}_R \left(|u_{rs}(\overline{w})|; k_2; \mu\right) \right\}$$

where $t_r(\overline{w}) = \overline{w}_r^T \nabla E_r$ and $u_{rs}(\overline{w}) = \overline{w}_r - \overline{w}_s$; the elements of the vector $\nabla E \equiv [E_x; E_y; E_t]^T$ represent the partial derivatives of a pair of images with respect to x, y, and t, respectively. The elements u and v of the vector field $\overline{w} = [u; v; 1]^T$ represent the component of the OF in the x and y directions, respectively. Figs. 8–a and 8–b show a synthetic image pair with a small square moving down and to the right. The computed OF field with quadratic potentials is shown in 8–c, and the OF computed with the presented technique in Fig. 8–d.

6 CONCLUSIONS

We have presented in this paper a "Quasi–robust" potential (QR) for edge–preserving regularization, that is formed by a convex combination of a robust (hard–redescending) potential and a quadratic one. We have shown that this allows one to control both the edge preserving and the granularity of the solution in a more accurate way. We have also presented an improved minimization algorithm for the corresponding cost function (Half-quadratic conjugated gradient), that permits one to obtain restorations that are of better quality that those obtained with other algorithms considered as the state of the art at a fraction of the computational cost. In the experiments presented here, we have used the QR potentials only in the regularization term; it is also possible however, to use them in the data term ρ_D to further improve the solution, particularly for high noise levels.

The added quadratic term acts as an stabilizer (regularization term) of the conjugated gradient algorithm avoiding the ill–conditioning of the resulting non–linear system because the weight functions (for the data and regularization terms) cannot be both zero (or very small). The combination of the QR potentials and the HQCG algorithm allows one to have a more graceful evolution of the solution at each iteration.

From the experiments we have performed, we have noted that the HQCG algorithm has the nice property of being "continuous" with respect the parameters (including the initial conditions), in the sense that its evolution is smooth (see Fig. 7), which allows one to observe this evolution and stop by hand when a solution that is perceptually "optimal" is reached even when the algorithm has not converged (see Fig. 6)

Acknowledgement 3 The authors thank to CONACYT (Mexico) for founding.

References

[1] S.Z. Li, "Markov Random Field Modeling in Computer Vision," Springer Verlag, 1995.

[2] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions and Bayesian restoration of images," IEEE Trans. Pattern Anal. Machine Intell., vol 6, 721-741, 1984.

[3] S. Geman and D.E. McClaure, "Bayesian image analysis methods: An application to single photon emission tomography," Proc. Statistical Computation Section, Amer. Statistical Assoc., Washington, DC, pp 12–18, 1985

[4] D. Geman and G. Reynolds, "Constrained restoration and the recovery of discontinuities," IEEE Trans. Image Processing, vol. 14, pp 367–383, 1992

[5] A. Rangarajan and R. Chellappa, "A continuation method for image restoration using the adiabatic approximation," in Markov random Fields: Theory and Applications, R. Chellappa and A.K. Jain (eds.) Academic Press Inc., 69–91, 1993.

[6] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," IEEE Trans. Image Processing, vol. 4, pp 932–946, 1995

[7] M.J. Black and A. Rangarajan, "Unification of line process, outlier rejection, and robust statistics with application in early vision," Int. J. Comput. Vis., vol. 19, no. 1, 57-92 1996.

[8] P. Charbonnier, L. Blanc-Féraud, G. Aubert and M. Barlaud, "Deterministic edge–preserving regularization in computer imagining," IEEE Trans. Image Processing, vol. 6, 298-311, 1997.

[9] M.J. Black, G. Sapiro, D.H. Marimont and D. Heeger, "Robust anisotropic diffusion," IEEE Trans. Image Processing, vol 7, 421-432, 1998.

[10] S. Teboul, L. Blanc-Féraud, G. Aubert and M. Barlaud, "Variational approach for edge–preserving regularization using coupled PDE's," IEEE Trans. Image Processing, vol. 7, 387–397,1998.

[11] T. Kubota and T, Huntsberg, "Adaptive anisotropic parameter estimation in the weak membrane model," in Proc. Int WS in EMMCVPR'97, Lecture Notes in Computer Vision 1223, Springer Verlag, Venice Italy, 179–194, 1997.

[12] Peter J. Huber, "Robust Statistics," Wiley, New York, 1981.

[13] G. Li "Robust Regression," in Exploring Data Tables, Trends, and Shapes, D.C. Hoaglin, F Mosteller, J.W. Tukey (eds.), 281–343, Wiley, New York, 1985.

[14] A. Blake and . Zisserman, "Visual reconstruction," The MIT Press, Cambridge, Massachusetts, 1987.

[15] J.E. Dennis and R.E. Welsch, "Techniques for nonlinear least squares and robust regression," Proc. Amer. Statis. Ass., 83–87, 1976.

[16] F.R.Hampel E.M. Ronchetti P.J. Rousseeuw and W.A. Stahel, "Robust Statistics: The approach based on Influence Functions," Wiley, New York, 1986.

[17] C.V. Stewart, "Robust parameter estimation in computer vision," SIAM Review, 41, 513–537, 1999.

[18] M.J. Black, D.J. Fleet and Y. Yacoob, "Robusty estimating changes in image appearance," Comput. Vision Image Understand. 78, 8–31, 2000.

[19] M. Ben–Ezra, S. Peleg and M. Werman "Real–time motion analysis with linear programming," Comput. Vision Image Understand. 78, 32–52, 2000.

[20] S.-H Lai "Robust Image matching under partial occlusion and spatial varying illumination changes," Comput. Vision Image Understand. 78, 84–78, 2000.

[21] P. Hellier, C. Barillot, É. Mémin, P. Pérez, "An energy–based framework for dense 3D registration of volumetric brain images," in Proc. Computer Vision and Pattern Recongnition (CVPR–00), vol 2, 270–275, Hilton Head Is. SC., 2000.

[22] W.J.J. Rey, "Introduction to Robust and Quasi–Robust Statistical Methods," Springer–Verlag, Berlin, Heidelberg, 1983.

[23] P.E. Gill, W. Murray and M.H. Wright, "Practical Optimization," Academic Press London, New York, 1981.

[24] C.T. Kelley, "Iterative Methods for Linear and Nonlinear Equations," Frontiers in Applied Mathemathatics, vol 16, SIAM, Philadelphia, 1995.

[25] J.J. Moré and D.J. Thuente, "Line search algorithms with guaranteed sufficient decrease," ACM Trans. Math. Softw. vol 20, 286–307, 1994.

[26] J.C. Gilbert and J. Nocedal,"Global convergence properties of conjugated gradient methods for optimization," SIAM J. Optim. vol 2, 21–42, 1992.

[27] P.K. Horn and B.G. Shunck, "Determining optical flow," Artificielle Intell., vol 17, 185–203, 1981.

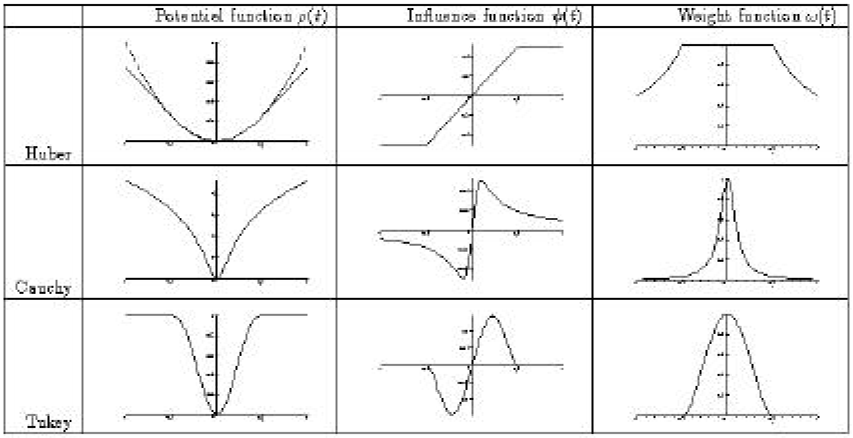

Figure 1: Examples of potentials with monotone, soft–redescending and hard–redescending in‡uence functions.

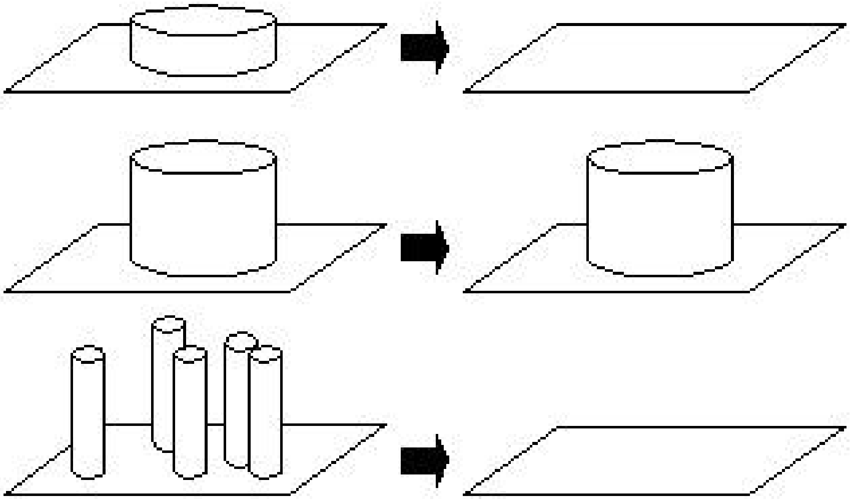

Figure 2: Desired behavior of the reconstruction algorithm: large regions with well–de…ned edges (middle) are preserved, but small regions are not (bottom)

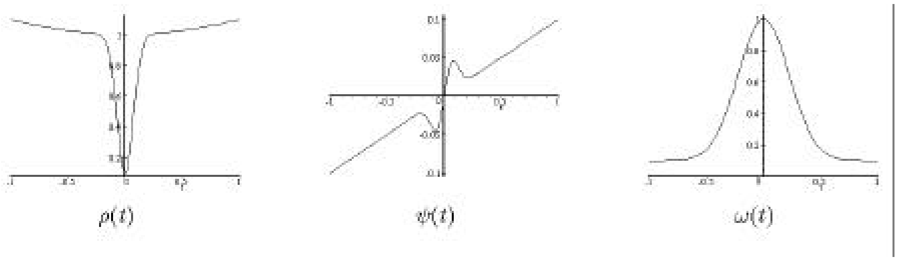

Figure 3: Potential ½$_{Q}$, in‡uence Ã and weight ! functions with explicit control over the de…nition and granularity

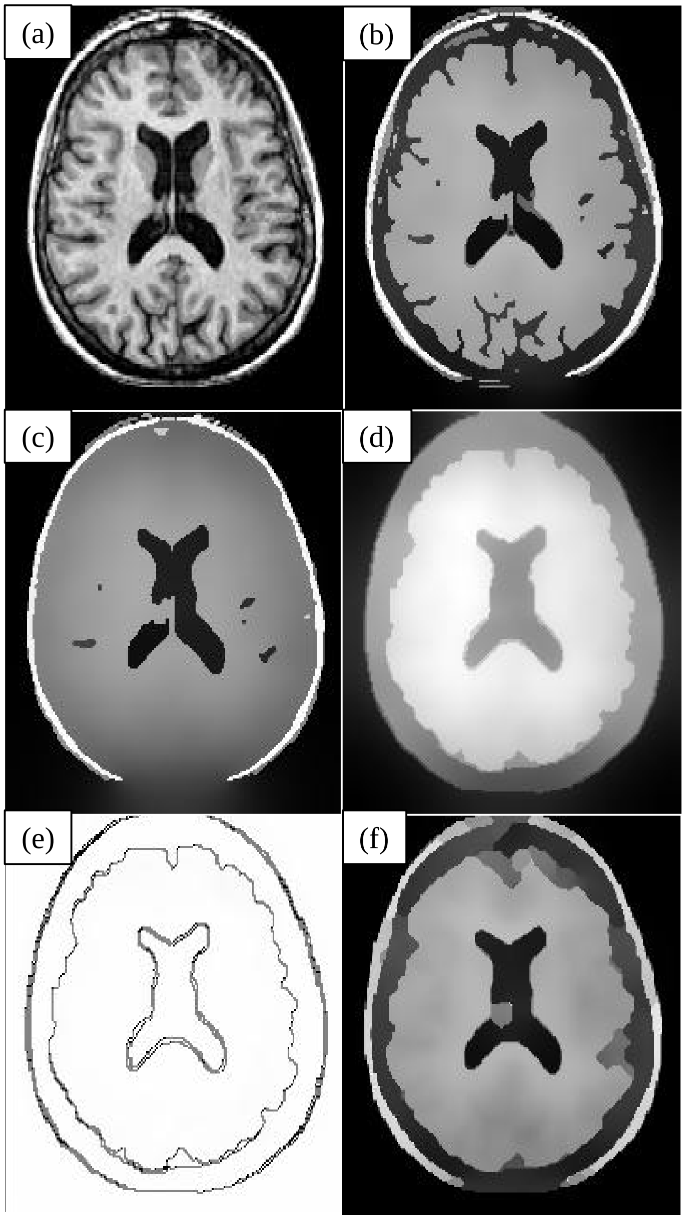

Figure 4: Segmentation of brain/non–brain tissue.(a) axial section of a MRI. (b) Restoration with HQGN with maximum granularity control (k_2 = 80 and ¹ = 0). (c) Restoration with HQGN algorithm with k_2 is decreased (k_2 = 60 and ¹ = 0). (d) Controlling granularity. Restoration with HQGN algorithm with k_2 = 800 and ¹ = 0:03. (e) Boundaries in panel e. (f) Filtered image with the ARTUR algorithm using the Geman–McClaude potential.

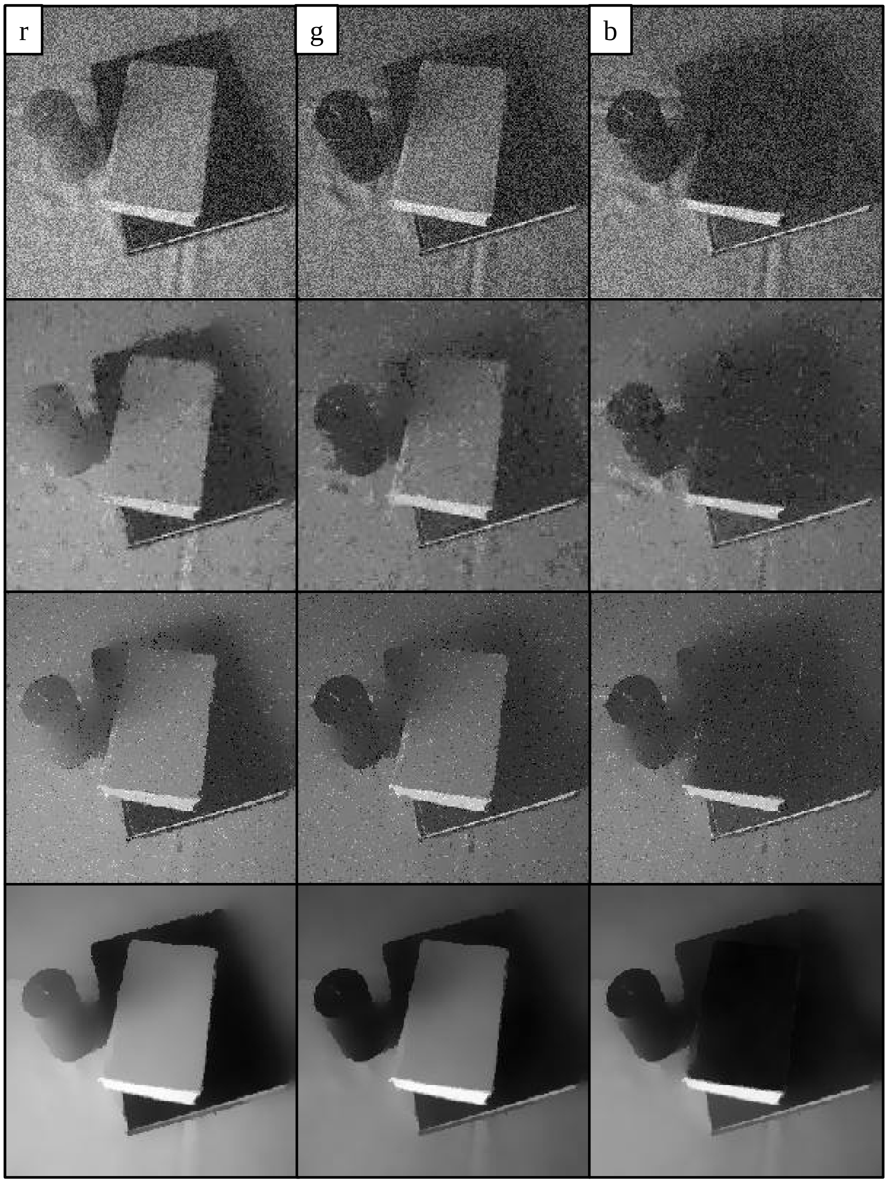

Figure 5: Vectorial valued image processing (channels red, green and blue of a color image). First row, noisy data. Second row, independently computed channels with a HR–potential. Third row, coupled computed channels with a HR–potential. Fourth row, coupled computed channels with a HR–potential and granularity constrain.

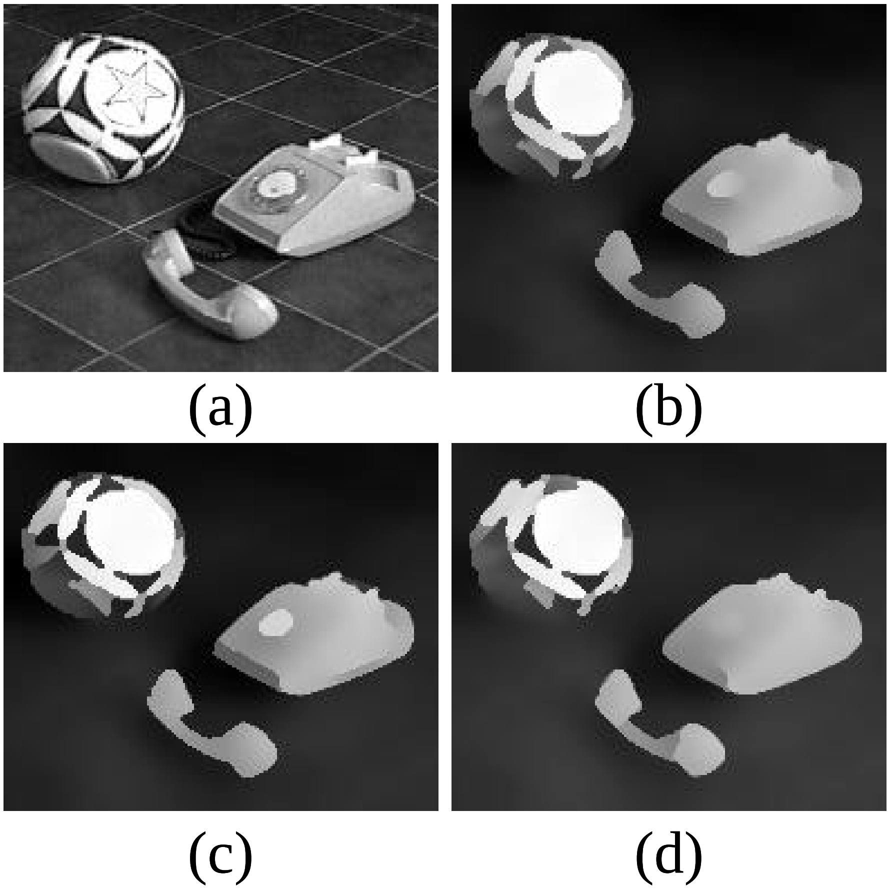

Figure 6: (a) Real image. Filtered with HQCG after (b) 30 iterations (6 secs.) and (c) 70 iterations. (14 secs.) (d) Filtered with ARTUR after 300 iterations (40 secs.).

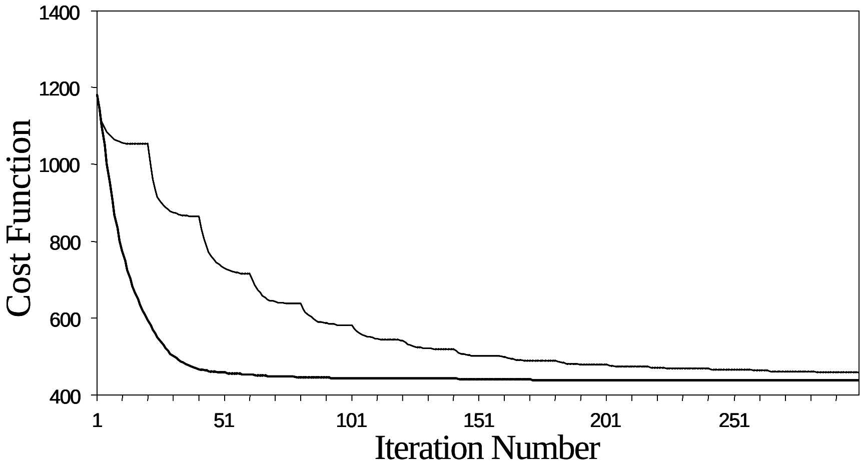

Figure 7: Preformance of the algorithms HQCG (bold line) and ARTUR–CG. Values of the cost function versus iteration number.

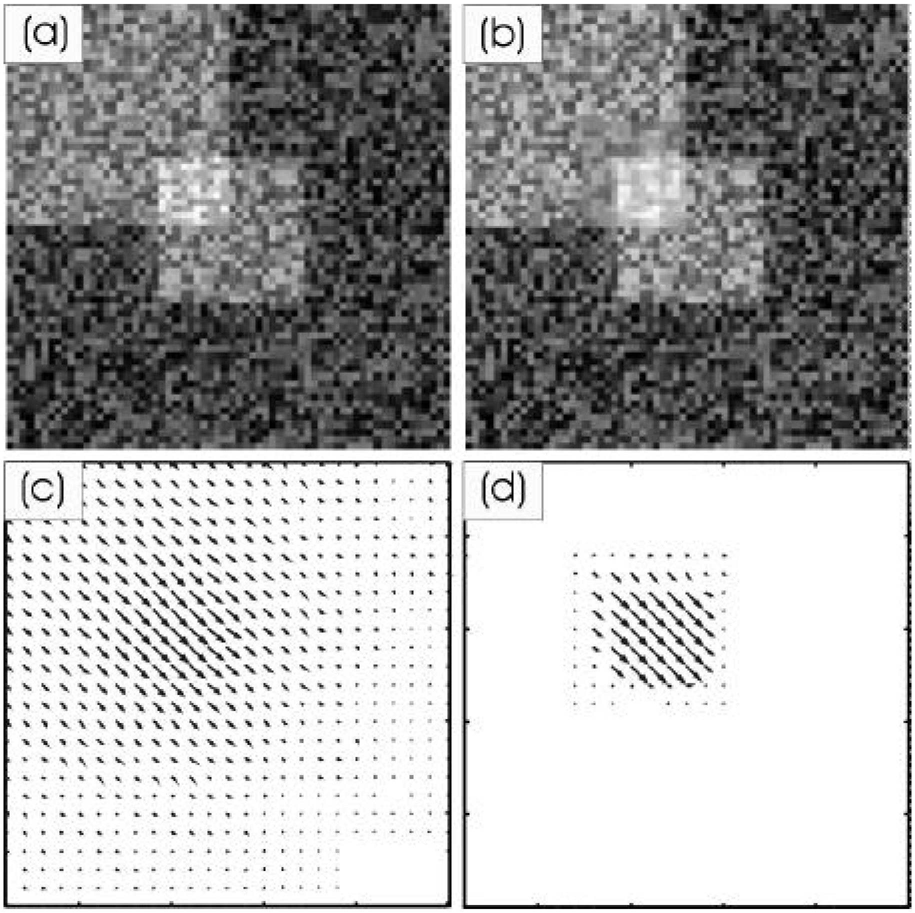

Figure 8: Optical ‡ow. (a) and (b) synthetic image pair with moving square. (c) Computed OF with quadratic potentials and (d) with robust potentials.