Communications in Applied and Industrial Mathematics, ISSN 2038-0909, 1, 2, (2010) 249–268
DOI: 10.1685/2010CAIM584

A deterministic algorithm for optical flow estimation

Ivan Gerace[1], Francesca Martinelli[2], Patrizia Pucci[1]

[1] *Dipartimento di Matematica e Informatica*
Università degli Studi di Perugia
Via Vanvitelli 1, I–06123 Perugia, Italy
gerace@dmi.unipg.it
pucci@dmi.unipg.it

[2] *Istituto di Scienza e Tecnologie dell'Informazione - CNR*
Via G. Moruzzi 1 56124 PISA - Italy
francesca.martinelli@isti.cnr.it

Communicated by Elena De Angelis

Abstract

In this paper we propose a new deterministic algorithm for determining optical flow through regularization techniques so that the solution of the problem is defined as the minimum of an appropriate energy function. We also assume that the displacements are piecewise continuous and that the discontinuities are variable to be estimated. More precisely, we introduce a hierarchical three–step optimization strategy to minimize the constructed energy function, which is not convex. In the first step we find a suitable initial guess of the displacements field by a gradient–based GNC algorithm. In the second step we define the local energy of a displacement field as the energy function obtained by fixing all the field with the exception of a row or of a column. Then, through an application of the shortest path technique we minimize iteratively each local energy function restricted to a row or to a column until we arrive at a fixed point. In the last step we use again a GNC algorithm to recover a sub–pixel accuracy. The experimental results confirm the goodness of this technique.

1. Introduction

Motion computation is a fundamental and difficult problem of *Computer Vision* which is concerned with either the computation of 3-D motion in the image space or the computation of 2-D motion in the image plane. In this paper, we deal with the latter problem, which is also called *optical flow* [1]. Optical flow is the distribution of apparent velocities of brightness patterns in a sequence of two or more images. The main common assumption used by different authors is the constancy intensity hypothesis of a

Received 2010 03 02, in final form 2011 03 01
Published 2011 03 12

pixel during its movement, that is the intensity of a point remains constant along its trajectory. We refer to such an assumption as the OFC (*Optical Flow Constraint*). Optical flow estimation methods can be classified into two main categories: *matching–based* and *gradient–based*. Matching–based methods make direct use of the OFC. They can handle large motion and avoid tricky derivative calculations, but they often meet with computational difficulties and yield poor sub–pixel accuracy. Gradient–based methods use a first order approximation of the OFC, and even if they have a relatively low computational cost and a good sub–pixel accuracy, they strongly fail in the case of multi–pixel displacements.

Like many other problems in the field of Computer Vision, optical flow is an *ill–posed inverse problem* in the sense of Hadamard [2]. This means that the existence, uniqueness and stability of the solution cannot be guaranteed. To regularize ill–posed problems, that is to select a single solution, which is robust with respect to the noise, the class of feasible solutions is restricted by imposing suitable constraints, which give additional information on the solution. Through regularization theory, a wide range of ill–posed problems can be reformulated in terms of variational principles [3]. The solution can thus be computed by optimizing suitable cost functionals that determine an implicit model for the solution. Such a model favors solutions that maximize measurements of smoothness. The piecewise smooth models are particularly interesting as they are applicable to general scenes. In fact, different objects in the image may have a different motion field and this causes discontinuity in the global field. Intensity edges play a primary role in guiding the search for discontinuities of other visual processes [4–6]. The reason for the critical role of intensity edges is intuitively clear. Indeed, returning to the optical flow problem, in several cases the locations of the discontinuities of the displacements correspond to the locations of the intensity edges. In general, displacement discontinuities are a subset of intensity edges, and only under specific circumstances motion discontinuity does not correspond to an edge intensity. Moreover, pixels which constitute both vertical and horizontal intensity edges, that is corners, have a primary importance to locate and compute the motion. Nagel proved that the motion constraint equation has a unique and analytical solution in correspondence to corner points [5,7]. Moreover, the correct estimation of the discontinuities of the optical flow is an indispensable building block for a more complex motion analysis [8]. Unfortunately, in this regularization approach the cost functional to minimize is usually not convex. This is due both to the fact that the problem is not linear and to the piecewise smoothness assumption [9]. Thus, minimization algorithms that can avoid local minima should be used. Stochastic relaxation algorithms, as simulated annealing, have the

DOI: 10.1685/2010CAIM584

desirable feature of asymptotically converging to a global minimum of any non– convex function [10]. However, they are computationally very expensive [11]. In [9] a less expensive sub–optimal method has been proposed. It is based on constructing a finite family of approximating cost functionals, in which the first is convex and the last is the original functional. The algorithm consists of minimizing each functional of the family one after the other, by an ordinary deterministic descent algorithm, starting form the local minimum reached at the previous iterate. This is the essence of the GNC (*Graduated Non–Convexity*) algorithm, firstly proposed in [12] for the image reconstruction.

In [11,13] a new deterministic approach is proposed, in which the optical flow estimation is reduced to the problem of finding the shortest path between two nodes of a directed graph. It is well–known that the shortest path is solvable in polynomial time. Unfortunately, in the case of two–dimensional images, the number of nodes necessary for the reduction is exponential with respect to the size of the images. To overcome this drawback in [11,13] is proposed to exclude, during the process, paths with excessive partial costs. This algorithm is also applicable to the image restoration problem [13]. Independently, in [14] a shortest path technique was introduced for the tracking deformable templates problem.

In this paper we propose a *hierarchical* three– step optimization strategy to minimize the energy function. In the first step we find a suitable initial guess of the displacements field by a gradient–based GNC algorithm. In this stage just piecewise smoothness constraints are considered. In the second step we define the local energy of a displacement field as the energy function obtained by fixing all the field with the exception of a row or a column. Then, after an application of the shortest path technique, we minimize iteratively each local energy function, restricted to a row or of a column, until we arrive at a fixed point. In this case the number of nodes required for the reduction to the shortest path problem is polynomial with respect to the size of a row or of a column. A similar approach has been proposed for the image restoration problem in [15]. Moreover, using such a technique it is fairly natural to consider a matching–based model and to introduce new constraints based on the texture structure and on the occlusion phenomena. Anyway, for this approach it is necessary to quantize the displacement field. Hence, in the last step we use again a GNC algorithm to recover a sub–pixel accuracy.

2. Problem of optical flow estimation

We consider a $n \times n$ color image $\mathscr{I}$ as the $n \times 3n$ matrix $\mathscr{I} = [\mathscr{I}^1, \mathscr{I}^2, \mathscr{I}^3]$ of three $n \times n$ matrices, $\mathscr{I}^k$, $k = 1, 2, 3$, where $\mathscr{I}^1$ is the $n \times n$ matrix of the red levels of the image, $\mathscr{I}^2$ is the $n \times n$ matrix of the blue levels and $\mathscr{I}^3$ is the $n \times n$ matrix of the green levels. Each entry of $\mathscr{I}$ belongs to $\mathbb{N} \cap [0, 255]$.

Let $\mathscr{I}_1 = [\mathscr{I}_1^1, \mathscr{I}_1^2, \mathscr{I}_1^3]$ and $\mathscr{I}_2 = [\mathscr{I}_2^1, \mathscr{I}_2^2, \mathscr{I}_2^3]$ be two different $n \times n$ color images of the same scene taken at different times. The estimation of optical flow consists in computing the displacement of each pixel from the first image to the second one. Assuming the classical OFC (*Optical Flow Constraints*), that is brightness does not change during the motion, we have to calculate the $n \times n$ matrix $D = [\boldsymbol{d}_{i,j}]$ of the displacements $\boldsymbol{d}_{i,j} = (u_{i,j}, v_{i,j})$, such that

$$\mathscr{I}_1^k(i, j) = \mathscr{I}_2^k(i + u_{i,j}, j + v_{i,j}), \;\; 1 \le i, j \le n, \, k = 1, 2, 3.$$

2.1. *Piecewise smoothness models*

It is well–known that the optical flow estimation problem is ill–posed in the sense of Hadamard [2]. In order to solve visual ill–posed problem, several approaches were investigate, like regularization, Bayesian and variational approaches [10,16]. In [9,11,13] some auxiliary variable associated to discontinuities of the displacement field are introduced. A good estimate of the discontinuities improves the quality of the results.

Definition 2.1. A *clique* c is a couple of points on a squared grid adjacent in a vertical or horizontal way. The symbol C indicates the set of all the cliques. Namely

$$C = \{c = \{(i, j), (h, l)\} : \;\; (i = h, j = l + 1) \vee (i = h + 1, j = l)\}.$$

Definition 2.2. A *line variable* b_c is a Boolean variable associated to each clique c. If a discontinuity in the displacement field is present on a clique c, then b_c assumes value one otherwise zero. The set $\boldsymbol{b}$ of all the line variables is called *line process.*

Definition 2.3. The *finite difference operator* $\Delta_c D$ *at any clique* c *of the displacement field* D is defined by

$$\Delta_c D = \sqrt{|u_{i,j} - u_{h,l}|^2 + |v_{i,j} - v_{h,l}|^2}.$$

DOI: 10.1685/2010CAIM584

By regularization techniques the solution of the optical flow estimation problem can be defined as the minimum of the following function called *primary energy function*:

$$(1)\qquad \begin{aligned} E(D,B) = &\sum_{i,j=1}^{n}\sum_{k=1}^{3}|\mathscr{I}_1^k(i,j) - \mathscr{I}_2^k(i+u_{i,j}, j+v_{i,j})|^2 \\ &+\sum_{c\in C}\left\{\lambda^2(\Delta_c D)^2(1-b_c)+\alpha b_c\right\}. \end{aligned}$$

The first term of (1) measures faithfulness of the solution to data. The second one, called *regularization term*, imposes a smoothness condition on D. The parameter λ^2 is a *regularization parameter*, that is a weight that balances, in the solution, the fitting to the data and the fidelity to the a priori information. The parameter α is positive and is used in order to avoid the presence of too many discontinuities in the displacement field D.

It is possible to minimize the primal energy function firstly with respect to all line variables b_c. The obtained *dual energy* function is defined as follows:

$$E_d(D) = \inf_B E(D,B).$$

Namely, $E_d(D)$ is the form :

$$(2)\quad E_d(D) = \sum_{i,j=1}^{n}\sum_{k=1}^{3}|\mathscr{I}_1^k(i,j) - \mathscr{I}_2^k(i+u_{i,j,k}, j+v_{i,j})|^2 + \sum_{c\in C}\phi(\Delta_c D),$$

where

$$(3)\qquad \phi(t) = \min\{\lambda^2 t^2, \alpha\} = \begin{cases} \lambda^2 t^2, & \text{if } |t| < s \\ \alpha & \text{otherwise,} \end{cases}$$

and λ^2 and α are fixed positive rational numbers. The quantity $s = \sqrt{\alpha}/\lambda$ has the meaning of a *threshold* for creating a discontinuity in the displacement field.

2.2. *Gradient–based approach*

The not global convexity of the energy is due both to not convexity of the ϕ, and to the fact that the convexity of the images $\mathscr{I}_1$, the image at the initial time $t = 1$ and $\mathscr{I}_2$, the evolved image at the final instant $t = 2$, is a priori not predictable. This fact is a peculiarity of the optical flow estimation, that distinguishes it from the image reconstruction problem. A way to overcome this problem we take for $\mathscr{I}_2^k(i,j)$ at the final instant $t=2$ the Taylor series of $\mathscr{I}_{1+\delta(1)}^k$, stopped at the first term and centered at (i,j), in the variable displacement $(u_{i,j}, v_{i,j})$ at the instant $1+\delta(1)$, where $\delta(1) = 1$ represents the time occurred from the acquisition of first image to the acquisition of the second or last image [1]. Therefore $u_{i,j} = \dot{u}_{i,j}\delta(1) = \dot{u}_{i,j}$ and $v_{i,j} = \dot{v}_{i,j}\delta(1) = \dot{v}_{i,j}$, that is the displacement and the velocity are the same with this normalization. Clearly the use of the unit time will give not the best approximation. In particular, for each $k=1,2,3$

$$\mathscr{I}_2^k(i+u_{i,j}, j+v_{i,j}) = \mathscr{I}_{1+\delta(1)}^k(i+u_{i,j}\delta(1), j+v_{i,j}\delta(1)) = \mathscr{I}_1^k(i,j).$$

Furthermore, the term on the left hand side, which appears in (1) and so in (2), is replaced by the first term of the Taylor formula, that is for $k=1,2,3$

$$\mathscr{I}_2^k(i+u_{i,j}, j+v_{i,j}) \doteq \mathscr{I}_1^k(i,j) + \partial_i \mathscr{I}_1^k(i,j)u_{i,j} + \partial_j \mathscr{I}_1^k(i,j)v_{i,j} + \partial_t \mathscr{I}_1^k(i,j).$$

Finally, an approximation of (2) is the following energy

$$\begin{aligned}\bar{E}_d(D) =& \sum_{i,j=1}^{n} \sum_{k=1}^{3} |\partial_i \mathscr{I}_1^k(i,j)u(i,j) + \partial_j \mathscr{I}_1^k(i,j)v(i,j) + \partial_t \mathscr{I}_1^k(i,j)|^2 \\ &+ \sum_{c\in C} \phi(\Delta_c D).\end{aligned} \tag{4}$$

This approximation can be considered accurate only if the displacements are very small. The choice $\delta(1) = 1$ is not always the best possible, but the unit discretization of the time is the most popular choice in optical flow estimation. The first term of $\bar{E}_d(D)$ in (4) is a convex function in D and this fact can, in a first approximation, help the estimation of the optical flow.

We recall that the constraint

$$\partial_i \mathscr{I}_1^k(i,j)u(i,j) + \partial_j \mathscr{I}_1^k(i,j)v(i,j) + \partial_t \mathscr{I}_1^k(i,j) = 0$$

for each $k=1,2,3$, is called MCE *Motion Constraints Equation*.

The minimization of (4) is the main aim of the gradient–based method.

2.3. *The rule of the intensity between discontinuities*

Many papers support the idea that intensity edges can be used as the primary cue to help location of discontinuities in the order processes such depth, motion and texture. This is due to the fact that discontinuities in depth, motion, texture typically yield large gradients in the image intensity.

DOI: 10.1685/2010CAIM584

It is sufficient to look around to convince ourselves that in the real world most of motion discontinuities, which occur at an intensity edge, can be better localized than motion, depth, texture. In an analogous way for the displacement field we can give the following

Definition 2.4. The *finite difference operator* $\Delta_c \mathscr{I}$ *at any clique* $c = \{(i,j),(h,l)\} \in C$ *of the intensity field* $\mathscr{I}$ is defined by

$$\Delta_c \mathscr{I} = \sqrt{\sum_{k=1}^{3} |\mathscr{I}^k(i,j) - \mathscr{I}^k(h,l)|^2}.$$

If, for fixed $\mathscr{I}$, the value of $\Delta_c \mathscr{I}$ is large, say $\Delta_c \mathscr{I} > S$, where S is *the intensity threshold*, this means that at the clique c *the intensity field presents a discontinuity.*

Using the information of the intensity field now we can change the iterating function (3) by

$$\text{(5)} \qquad \Phi(\Delta_c D, \Delta_c \mathscr{I}_1) = \begin{cases} |\Delta_c D|^2 & \text{if } |\Delta_c \mathscr{I}_1| \leq S, \\ \phi(\Delta_c D) & \text{if } |\Delta_c \mathscr{I}_1|\, S, \end{cases}$$

where ϕ in (3) takes into account of the discontinuities of the displacement field and $\mathscr{I}_1$ is the first image. In this way every discontinuity of the displacement field at the clique c is also a discontinuity of the intensity field. Let now call *corner* a particular configuration of the discontinuities of the intensity field which occurs in both sides. The corners are classified in four types as shown in Figure 1.

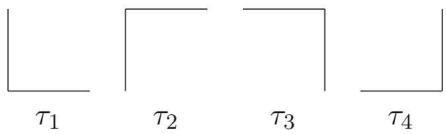


Fig. 1. Four types of corners

Let $f(i,j,\mathscr{I},\tau)$ be a Boolean function, which is equal to *true* if and only if the pixel (i,j) is at a corner of the type τ of the image $\mathscr{I}$, see the corners classification in Figure 1. Let us consider now the function

$$\text{(6)} \quad \Psi = \Psi(i,j,\boldsymbol{d}_{i,j},\mathscr{I}_1,\mathscr{I}_2) = \begin{cases} -\gamma & \text{if there exists } \tau \in \{\tau_1,\tau_2,\tau_3,\tau_4\} \\ & \text{such that } \big\{f(i,j,\mathscr{I}_1,\tau) \\ & \quad \wedge f(i+u_{i,j}, j+v_{i,j}, \mathscr{I}_2, \tau)\big\}, \\ 0 & \text{otherwise,} \end{cases}$$

where γ is a positive rational number. The function Ψ in (6) gives a reward when a corner of the first image $\mathscr{I}_1$ is moved into a corner of the second image $\mathscr{I}_2$.

Taking into account of (5) and (6) the main dual energy function in (2) takes the final form

$$
\begin{aligned}
E_d(D) = \sum_{i,j=1}^{3} \Big\{ & \sum_{k=1}^{3} |\mathscr{I}_1^k(i,j) - \mathscr{I}_2^k(i+u_{i,j}, j+v_{i,j})|^2 \\
& + \Psi(i,j,\boldsymbol{d}_{i,j},\mathscr{I}_1,\mathscr{I}_2) \Big\} + \sum_{c\in C} \Phi(\Delta_c D, \Delta_c \mathscr{I}_1).
\end{aligned} \tag{7}
$$

3. The minimization procedure

In order to minimize the energy function E_d in (7) we *first* obtain a good initial guess $D_1 = D_1^{(0)}$ by minimizing the energy function $\bar{E}_d$ in (4) from an initial point D_0. Actually, the standard procedure, largely used in the literature by the gradient–based approach, is to take D_1 as a final solution D of the original problem. However, this approach typically fails in case of large displacements, since a first order approximation is clearly inaccurate. *In the second step*, we minimize the energy function E_d in (7) starting from the initial guess $D_1 = D_1^{(0)}$ obtained by the gradient–based algorithm from an initial point D_0. The minimization is performed by a new deterministic iterative algorithm presented later, which uses only pairs of integer numbers for the entries $\boldsymbol{d}_{ij}$ of the displacements and converges to a fixed point D_2 in a finite number of iteration. Finally, *in the last step*, a refinement of the entries $\boldsymbol{d}_{ij}$ of the obtained fixed point D_2 is performed by another gradient–based algorithm and the output is the final result D of the minimization process.

3.1. *The first step via a gradient–based GNC algorithm*

The optical flow estimation solution D is usually obtained via a gradient–based approach by minimizing the energy function $\bar{E}_d$ in (4), and using a classical deterministic or stochastic technique, see [1,9,17]. In particular, in [9] the technique adopted is called GNC (*Graduated Non–Convexity*) algorithm, which is used to approximate the value of the global minimum of a non convex energy function as $\bar{E}_d$ and depends on the choice of the starting point. In order to have a suitable choice of this point, the GNC technique requires to find a finite family of approximating functions $\{E_d^{(p_\kappa)}\}_{\kappa\in\{1,\ldots,\bar{\kappa}\}}$, such that the first $E_d^{(p_1)}$ is convex and the last $E_d^{(p_{\bar{\kappa}})} = \bar{E}_d$ is the original dual energy function in (4). Hence we apply the following algorithm from an initial point D_0:

`Algorithm 1`
`Instance:` $(\mathscr{I}_1, \mathscr{I}_2, D_0)$
`Output:` $D_{0,\bar{\kappa}} = D_1$
    $\kappa = 1$;
    $D_{0,0} = D_0$;
    `while` $\kappa \neq \bar{\kappa}$ `do:`
      $D_{0,\kappa}$ `is equal to the stationary point among`
      `the speediest descent direction of` $E_d^{(p_\kappa)}$,
      `starting from` $D_{0,\kappa-1}$;
      $\kappa = \kappa + 1$;
    `end while`

Actually, the classical GNC algorithm, described above, was proposed in [12] for the image restoration problem. In this paper we use the GNC algorithm modified by Nikolova in [18], which can be applied to data terms which are not necessarily strictly convex as in our case. Indeed, the degree of convexity of the consistence data term of the function $\bar{E}_d$ in (4) is difficult to estimate.

To reach the stationary point $D_{0,\kappa}$ of the κ–th step of the GNC algorithm the NLSOR (*Non–Linear Successive Over– Relaxation*) is used, that is

`Algorithm 1.1`
`Instance:` $(\mathscr{I}_1, \mathscr{I}_2, D_{0,\kappa-1})$
`Output:` $D_{0,\kappa}$
    $\ell = 1$;
    $D_{0,\kappa-1}^{(0)} = D_{0,\kappa-1}$;
    `while, either the stationary point is not found`
    `or` $\ell = \bar{\ell}$, `do`
      $$D_{0,\kappa-1}^{(\ell)} = D_{0,\kappa-1}^{(\ell-1)} - \frac{\omega}{T}[\partial_{\boldsymbol{d}_{i,j}} E_d^{(p_\kappa)}(\mathscr{I}_1, \mathscr{I}_2, D_{0,\kappa-1}^{(\ell-1)})]_{i,j=1}^n;$$
      $\ell = \ell + 1$;
    `end while`

where $\bar{\ell}$ is the maximum number of iterations we decide to performance, $0 < \omega < 2$ is called *NLSOR parameter*, governing speed of convergence, and T is an upper bound on the second derivatives of $E_d^{(p_\kappa)}$, that is

$$\max_{i,j=1,\dots,n} |\partial^2_{\boldsymbol{d}^2_{i,j}} E_d^{(p_\kappa)}(\mathscr{I}_1, \mathscr{I}_2, D)| \leq T$$

for all $\mathscr{I}_1$, $\mathscr{I}_2$ and D.

3.2. *The second step via matching–based algorithm*

In order to minimize the energy function E_d in (7) we define the *local energy function along a row or along a column*. We first introduce and discuss the properties of the energy function along a row i, which is denoted in the following by $E_d^i(\boldsymbol{d}_i)$. Let $i \in \{1, 2, \ldots, n\}$ and

$$I_i(j) = \Big\{\{(i,j),(i+1,j)\},\{(i,j),(i-1,j)\},\{(i,j),(i,j-1)\}\Big\} \subset C.$$

The energy E_d^i along the i–th row $\boldsymbol{d}_i$ of the displacement field D is defined by

$$E_d^i(\boldsymbol{d}_i) = \sum_{j=1}^{n} \rho(i, j, \boldsymbol{d}_{i,j-1}, \boldsymbol{d}_{i,j}), \tag{8}$$

with

$$\begin{aligned}\rho(i, j, \boldsymbol{d}_{i,j-1}, \boldsymbol{d}_{i,j}) = {} & \sum_{k=1}^{3} \big|\mathscr{I}_1^k(i,j) - \mathscr{I}_2^k(i + u_{i,j}, j + v_{i,j})\big|^2 \\ & + \Psi(i, j, \boldsymbol{d}_{i,j}, \mathscr{I}_1, \mathscr{I}_2) + \sum_{c \in I_i(j)} \Phi(\Delta_c D, \Delta_c \mathscr{I}_1),\end{aligned}$$

where $\boldsymbol{d}_{i,0}$ is replaced in the following by the symbol $-$ in order to emphasize that there is not a previous pixel in the line i.

In a similar way we define the energy function E_d along a column j, denoted by $\mathscr{E}_d^j(\boldsymbol{\delta}_j)$, where $\boldsymbol{\delta}_j$ is the j–th column of D.

We assume that each entry of D belongs to

$$\Omega = \mathbb{N}^2 \bigcap \Big([-m, m] \times [-m, m]\Big) \bigcup \{\perp\}.$$

Thus the number M of possible displacements for each pixel (i, j), such that

$$\max\{|i-1|,\ |j-1|,\ |i-n|,\ |i-n|\} \leq m < n/2,$$

is given by

$$M = |\Omega| = (2m+1)^2 + 1. \tag{9}$$

Clearly in the practical interesting situations the parameter m is much smaller that $n/2$. The set Ω can be ordered and in this case it will be

DOI: 10.1685/2010CAIM584

denoted by $\{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ..., \boldsymbol{\omega}_M\}$.

The problem to estimate the optical flow along a line can be formalized in the following way:

Instance: Two images $\mathscr{I}_1, \mathscr{I}_2$ and a line index i.

Problem: Determine $\boldsymbol{d}_i \in \Omega^n$ that minimize the function $E_d^i(\boldsymbol{d}_i)$ in (8).

The problem is now reduced to determine the shortest path in a direct graph. Indeed, *a direct graph* $G = G(N, A)$ is defined as the couple consisting of a set N of nodes and of a set A of ordered couples of nodes. These later are called *arches*.

Let $a = (n_i, n_j) \in A$, we call the node n_i as *the entering node* of a and n_j *the leaving node* of a.

Two arches are called *contiguous* if the leaving node of the first coincides with the entering node of the latter.

A *path* $P = (a_1, a_2, ...a_k)$ from the node n_i to the node n_j is a ordered set of arches such that a_l and a_{l+1}, $l = 1, 2, ..., k-1$, are contiguous and that the entering node of a_1 is n_i and the leaving node a_k is n_j

Let $p : A \to \mathbb{R}$ be a function that gives a cost to every arches. The *cost* $\pi(P)$ *of a path* $P = (a_1, a_2, ..., a_k)$ is defined as

$$\pi(P) = \sum_{l=1}^{k} p(a_l). \tag{10}$$

The problem of the shortest path can be formalized in following way.

Instance: A direct graph $G = G(N, A)$, a cost function $p : A \to \mathbb{R}$ a source node n_i, a terminal node n_j.

Problem: Determine a path P from n_i to n_j such that its cost is minimum.

Theorem 3.1. *The problem of estimating the optical flow along a line can be polynomially reduced in a shortest path problem. That is the optical flow estimation problem along a line can be solved in polynomial time.*

Proof. In order to polynomially reduce a problem $\mathscr{A}$ to a problem $\mathscr{B}$ it is enough to prove that (i) every instance a of the problem $\mathscr{A}$ can be transformed in a polynomial time into an instance b_a of the problem $\mathscr{B}$; (ii) if $s(b_a)$ denotes the solution of b_a in $\mathscr{B}$, then $s(b_a)$ is transformed in a polynomial time into the optimum solution $s(a)$ of $\mathscr{A}$ corresponding to the initial instance a in (i).

Let $\Omega = \{\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, ..., \boldsymbol{\omega}_M\}$. In order to transform the instance $\mathscr{I}_1$, $\mathscr{I}_2$, i of the flow estimation problem we proceed as follows: first we construct a direct graph $G = G(N, A)$, with

$$\begin{aligned} N =&\Big\{(0,-),(1,\boldsymbol{\omega}_1),\ldots,(1,\boldsymbol{\omega}_M),(2,\boldsymbol{\omega}_1),\ldots,(2,\boldsymbol{\omega}_M),\ldots,\\ &(n-1,\boldsymbol{\omega}_1),\ldots,(n-1,\boldsymbol{\omega}_M),(n,\boldsymbol{\omega}_1),\ldots,(n,\boldsymbol{\omega}_M),(n+1,-)\Big\},\\ A =&\Big\{((j,\boldsymbol{\omega}_h),(j+1,\boldsymbol{\omega}_l))\,:\,j=1,...n-1,h=1,...,M,l=1,...,M\Big\}\cup\\ &\Big\{((0,-),(1,\boldsymbol{\omega}_h))\,:\,h=1,...,M\Big\}\cup\Big\{((n,\boldsymbol{\omega}_h),(n+1,-))\,:\,h=1,...,M\Big\}. \end{aligned}$$

The source and the terminal node are denoted respectively by $(0, -)$ and by $(n + 1, -)$ (see Figure 7). Then we define a cost function $p : A \to \mathbb{R}$ by

$$\begin{aligned} p((j,\boldsymbol{\omega}_h),(j+1,\boldsymbol{\omega}_l)) &= \rho(i,j,\boldsymbol{\omega}_h,\boldsymbol{\omega}_l), && j=1,\ldots,n-1,\ h,l=1,\ldots,M,\\ p((0,-),(1,\boldsymbol{\omega}_h)) &= \rho(i,1,-,\boldsymbol{\omega}_h), && h=1,\ldots,M,\\ p((n,\boldsymbol{\omega}_h),(n+1,-)) &= 0, && h=1,\ldots,M. \end{aligned}$$

It is easy to prove that such a reduction is polynomial time computable and property (i) is proved.

Let us now analyze the characteristic of the solution of the shortest path problem. ¿From the source node $(0, -)$ is possible to reach just nodes of type $(1, \boldsymbol{\varpi}_1)$, with $\boldsymbol{\varpi}_1 \in \Omega$. Analogously, from a node $(j, \boldsymbol{\varpi}_j)$, with $j = 1, \ldots, n-1$, $\boldsymbol{\varpi}_j \in \Omega$, it is possible to reach just nodes of type $(j+1, \boldsymbol{\varpi}_{j+1})$, with $\boldsymbol{\varpi}_{j+1} \in \Omega$. Finally, from a node of type $(n, \boldsymbol{\varpi}_n)$, with $\boldsymbol{\varpi}_n \in \Omega$, it is possible to reach only the terminal node $(n + 1, -)$. Thus a solution of the shortest path problem is given by

$$P = ((0,-),(1,\boldsymbol{\varpi}_1),\ldots,(n,\boldsymbol{\varpi}_n),(n+1,-))\,, \tag{12}$$

with $\boldsymbol{\varpi}_j \in \Omega$, $j = 1, \ldots, n$. The cost of the shortest path P is given by

$$\begin{aligned} \pi(P) &= p((0,-),(1,\boldsymbol{\varpi}_1)) + \sum_{j=2}^{n} p\left((j-1,\boldsymbol{\varpi}_{j-1}),(j,\boldsymbol{\varpi}_j)\right)\\ &\quad + p((n,\boldsymbol{\varpi}_n),(n+1,-))\\ &= \rho(i,1,-,\boldsymbol{\varpi}_1) + \sum_{j=2}^{n} \rho(i,j,\boldsymbol{\varpi}_{j-1},\boldsymbol{\varpi}_j). \end{aligned}$$

Thus this problem is equivalent to find the argument $(\boldsymbol{\varpi}_1, \ldots, \boldsymbol{\varpi}_n) \in \Omega^n$ of the minimum of the function

$$\rho(i,j,-,\boldsymbol{\varpi}_1) + \sum_{j=2}^{n} \rho(i,j,\boldsymbol{\varpi}_{j-1},\boldsymbol{\varpi}_j);$$

DOI: 10.1685/2010CAIM584

that is exactly to the problem of estimating the optical flow along a line i given in (8), with

$$\boldsymbol{d}_{i,j} = \boldsymbol{\varpi}_j, \qquad j = 1, ..., n. \tag{15}$$

Clearly the transformation (15) is polynomial time computable. This completes the proof of (ii), and of the first part of the theorem.

Finally, since the shortest path problem is solvable in polynomial time, by the polynomial reduction, constructed above, the problem of estimating the optical flow along a line is polynomial time computable. □

Using the reduction described in the proof of Theorem 3.1, the cost of the algorithm to compute the optical flow along a line turns to be $O(M^2 n^2)$. The matching–based algorithm proposed for the estimation of the optical flow in the entire displacement field, which starts from the output D_1 of Algorithm 1, is the following

```
Algorithm 2
Instance: (𝒥1, 𝒥2, D1)
Output: D2
   Current_Energy = Ed(D1)
   do:
     Old_Energy = Current_Energy;
     for i = 1 to n
       update the optical flow along the i-th row
       by the polynomial algorithm;
     for j = 1 to n
       update the optical flow along the j-th
       column by the polynomial algorithm;
     Current_Energy = Ed(D);
   until Current_Energy = Old_Energy
```

Theorem 3.2. *Assume that the positive parameters α, γ, λ^2 and ν are rational numbers. Then the algorithm above converges in a finite number N of steps.*

Proof. It is convenient to follow the process of iteration step by step. First we fix an initial displacement $D^{(0)} = D_1$ in Ω^{n^2}, then we evaluate

$$E_d(\boldsymbol{d}_1, \boldsymbol{d}_2^{(0)}, \dots, \boldsymbol{d}_n^{(0)}) = E_d^1(\boldsymbol{d}_1) + R_1(\boldsymbol{d}_2^{(0)}, \dots, \boldsymbol{d}_n^{(0)}),$$

and minimize $E_d^1(\boldsymbol{d}_1)$, when $\boldsymbol{d}_1$ varies in Ω^n, obtaining $\tilde{\boldsymbol{d}}_1$. Next we evaluate

$$E_d(\tilde{\boldsymbol{d}}_1, \boldsymbol{d}_2, \boldsymbol{d}_3^{(0)} \dots, \boldsymbol{d}_n^{(0)}) = E_d^2(\boldsymbol{d}_2) + R_2(\tilde{\boldsymbol{d}}_1, \boldsymbol{d}_3^{(0)}, \dots, \boldsymbol{d}_n^{(0)}),$$

and minimize $E_d^2(\boldsymbol{d}_2)$, when $\boldsymbol{d}_2$ varies in Ω^n, obtaining $\tilde{\boldsymbol{d}}_2$, and so on until we find the matrix $\tilde{D} = [\tilde{\boldsymbol{d}}_1, \ldots, \tilde{\boldsymbol{d}}_n]^T = [\tilde{\boldsymbol{\delta}}_1, \ldots, \tilde{\boldsymbol{\delta}}_n]$.

Successively, we evaluate

$$E_d(\boldsymbol{\delta}_1, \tilde{\boldsymbol{\delta}}_2, \ldots, \tilde{\boldsymbol{\delta}}_n) = \mathscr{E}_d^1(\boldsymbol{\delta}_1) + \mathscr{R}_1(\tilde{\boldsymbol{\delta}}_2, \ldots, \tilde{\boldsymbol{\delta}}_n),$$

and minimize $\mathscr{E}_d^1(\boldsymbol{\delta}_1)$, when $\boldsymbol{\delta}_1$ varies in Ω^n, obtaining $\boldsymbol{\delta}_1^{(1)}$; and so on until finding the matrix $D^{(1)} = [\boldsymbol{\delta}_1^{(1)}, \ldots, \boldsymbol{\delta}_n^{(1)}] = [\boldsymbol{d}_1^{(1)}, \ldots, \boldsymbol{d}_n^{(1)}]^T$.

In this way we get

$$E_d(D^{(1)}) = \varepsilon_1 \leq E_d(D^{(0)}) = \varepsilon_0,$$

by construction. If $\varepsilon_1 = \varepsilon_0$, the algorithm ends. Otherwise, $\varepsilon_0 - \varepsilon_1 \geq \varepsilon > 0$, where $\varepsilon > 0$ is the minimum variation of the energy given by

$$\varepsilon^{-1} = \text{m.c.m.}\{\alpha_d, \gamma_d, \lambda_d^2, \nu_d\},$$

if $\alpha = \alpha_n/\alpha_d$, $\gamma = \gamma_n/\gamma_d$, $\lambda^2 = \lambda_n^2/\lambda_d^2$ and $\nu = \nu_n/\nu_d$, with α_n, α_d, γ_n, γ_d, λ_n^2, λ_d^2, ν_n and ν_d in $\mathbb{N}^+$. Indeed,

$$\varepsilon^{-1} E_d(D) \in \mathbb{N} \quad \text{for all } D \in \Omega^{n^2},$$

being $\Omega^{n^2} \subset \mathbb{N}^{2n^2}$. We iterate the procedure so that the k–th step

$$E_d(D^{(k+1)}) = \varepsilon_{k+1} \leq E_d(D^{(k)}) = \varepsilon_k.$$

If $\varepsilon_{k+1} = \varepsilon_k$, the algorithm ends. Otherwise $\varepsilon_k - \varepsilon_{k+1} \geq \varepsilon$. Clearly the iteration ends in a finite number N of steps, with

$$N \leq \frac{\varepsilon_0 + \gamma n^2}{\varepsilon} \in \mathbb{N},$$

since

$$\min_{D \in \Omega^{n^2}} E_d(D) \geq \min E_d(D, \mathscr{I}, \mathscr{J}) = -\gamma n^2$$

and $\mathscr{I}_1$ and $\mathscr{I}_2$ in (7) are replaced by any images $\mathscr{I}$ and $\mathscr{J}$. □

3.3. *The third step via gradient–based algorithm*

Trough the second step of the proposed minimization procedure it is possible to find an integer estimation of the displacements quite close to the optimum one, anyway, in order to get the final output displacement D with real entries, we apply again the GNC algorithm. This algorithm uses the finite family of approximating functions of the Algorithm 1, that is, $\{E_d^{(p_\kappa)}\}_{\kappa \in \{1,\ldots,\bar{\kappa}\}}$, with the first $E_d^{(p_1)}$ convex and the last $E_d^{(p_{\bar{\kappa}})} = \bar{E}_d$

the original dual energy function in (4), but with slight differences. More specifically, we apply Algorithm 1 from the initial point D_2, output of Algorithm 2, as follows:

```
Algorithm 3
Instance: (𝒥_1, 𝒥_2, D_2)
Output: D_{2,κ̄} = D
   κ = 1;
   D_{2,0} = D_2;
   while κ ≠ κ̄ do:
     D_{2,κ} is equal to the stationary point among
     the speediest descent direction of E_d^{(p_κ)},
     starting from D_{2,κ-1};
     κ = κ + 1;
   end while
```

To reach the stationary point $D_{2,\kappa}$ of the κ–th step of the GNC Algorithm 3 again the NLSOR (*Non–Linear Successive Over– Relaxation*) is performed, but the maximum number of iterations $\hat{\ell}$ is now much smaller than the maximum number of iterations $\bar{\ell}$ used in Algorithm 1.1.

```
Algorithm 3.1
Instance: (𝒥_1, 𝒥_2, D_{2,κ-1})
Output: D_{2,κ}
   ℓ = 1;
   D_{2,κ-1}^{(0)} = D_{2,κ-1};
   while, either the stationary point is not found
   or ℓ = ℓ̄, do
     D_{2,κ-1}^{(ℓ)} = D_{2,κ-1}^{(ℓ-1)}
                  - (ω/T)[∂_{d_{i,j}} E_d^{(p_κ)}(𝒥_1, 𝒥_2, D_{2,κ-1}^{(ℓ-1)})]_{i,j=1}^n;
     ℓ = ℓ + 1;
   end while
```

where the involved parameters are introduced in Section 3.1.

We end, justifying that the main reason for the choice of the maximum number of iterations $\hat{\ell}$ as a small integer is due to fact that the output D does not lose in this way the good properties obtained in the construction of D_2.

4. Experimental results

The performance of the algorithm has been tested both on real and synthetic images. The intensity process has been quantized in 256 grey levels for grey–scale images, while in 256 levels for each of the three RGB canals used for color images. In the Algorithm 2 the displacements have been considered to have integer values, corresponding to multiples of pixels. In order to test the quality of the obtained displacement field, we have constructed a new image taking the pixels of the first image and moving them according to the displacement field, according to the formula

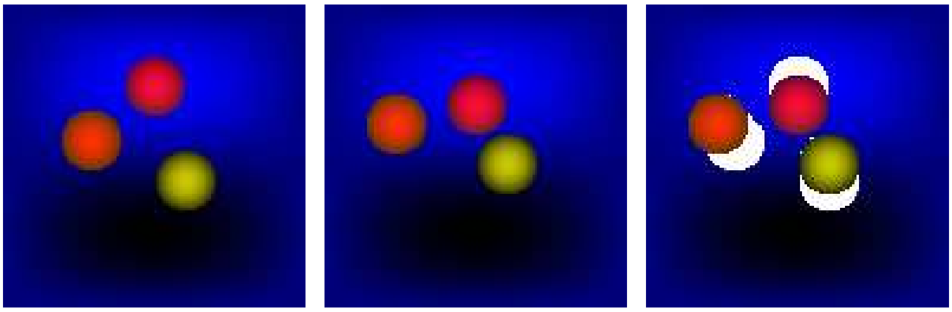
$$\mathscr{I}_3^k(i,j) = \mathscr{I}_2^k(i + \mathrm{round}(u_{i,j}), j + \mathrm{round}(v_{i,j})),$$

for $1 \leq i,j \leq n$ and $k = 1,2,3$. We refer to the image $\mathscr{I}_3$ as the *displaced image*, cf. the notation of Section 2. Note that due to the effect of occlusions in the displaced image some values of the pixels intensity are undefined, so we set them to the value white. In the examples below the choice of the involved free parameters α, λ is obtained experimentally. The correct estimation of these hyper-parameter is actually still an open problem. In this set of experimental results we have fixed $m = 10$ in (9). In a first experiment we used two synthetic grey–scale images of size 128×128, see Figure 2(a)–(b). In Figure 2(c) we present the obtained displaced image $\mathscr{I}_3$. In a second experiment we consider two synthetic color image of size 128×128 Figure 3(a)–(b). The displaced image between these two images, obtained with $\lambda = 16$ and $\alpha = 256$, is presented in Figure 3(c). The last two experiments are real and for them we do not present the obtained displaced image $\mathscr{I}_3$, since the main assumption OFC, cf. Section 1, is usually too strong in real cases. This is due to the fact that frequently is the light source that moves instead of the pixels. The first example is given in Figure 4(a)–(b) and the two real images are of size 180×180. Figure 5 shows the displacement field D obtained by the hierarchical three–step optimization algorithm, with $\lambda = 50$ and $\alpha = 7500$. The latter real example is presented in Figure 6(a)–(b), where now the two real images are of size 240×240, while the displacement field D, obtained by the hierarchical three–step optimization algorithm, with $\lambda = 8$ and $\alpha = 320000$ is given in Figure 6(c). Finally we compared the results of the proposed algorithm with the results of different simulated annealing scheme, but we noticed that to obtain the same amount of final energy function for the stochastic algorithm was necessary at least seven times the computational time of our deterministic algorithm.

DOI: 10.1685/2010CAIM584

(a) (b) (c)

Fig. 2. (a) First image; (b) Second image; (c) The displaced image obtained with $\lambda = 4$ and $\alpha = 16$



(a) (b) (c)

Fig. 3. (a) First image; (b) Second image; (c) The displaced image obtained with $\lambda = 16$ and $\alpha = 256$

(a) (b)

Fig. 4. (a) First image; (b) second image

5. Conclusions

In this paper, we address the problem of optical flow estimation. The solution is defined as a minimum of a suitable energy function, which is

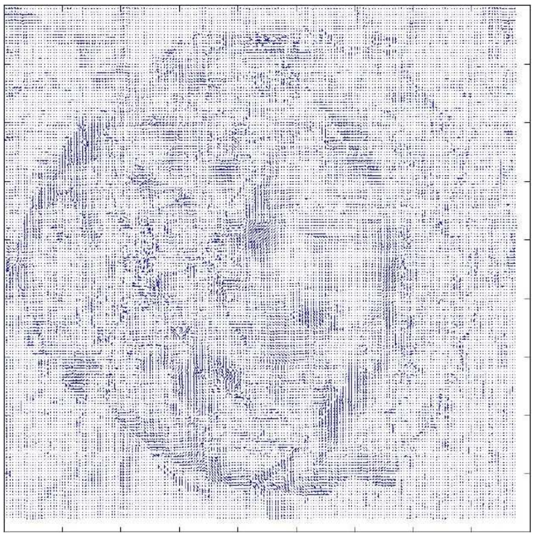


Fig. 5. Displacement field between the image in Figure 4 (a) and 4 (b) obtained with $\lambda = 50$ and $\alpha = 7500$;

(a) (b) (c)

Fig. 6. (a) First image; (b) second image; (c) displaced image obtained with $\lambda = 8$ and $\alpha = 320000$

not convex. The displacements are assumed to be piecewise continuous and their discontinuities are variable to be estimated. More precisely, a hierarchical three–step optimization strategy to minimize the constructed energy function, which is not convex, is presented and seems new. In the first step a suitable initial guess of the displacements field is chosen by a gradient–based GNC algorithm. In the second step the local energy of a displacement field is defined as the energy function obtained by fixing all the field with the exception of a row or of a column. Then, through an application of the shortest path technique we minimize iteratively each local energy function restricted to a row or to a column until we arrive at a fixed

DOI: 10.1685/2010CAIM584

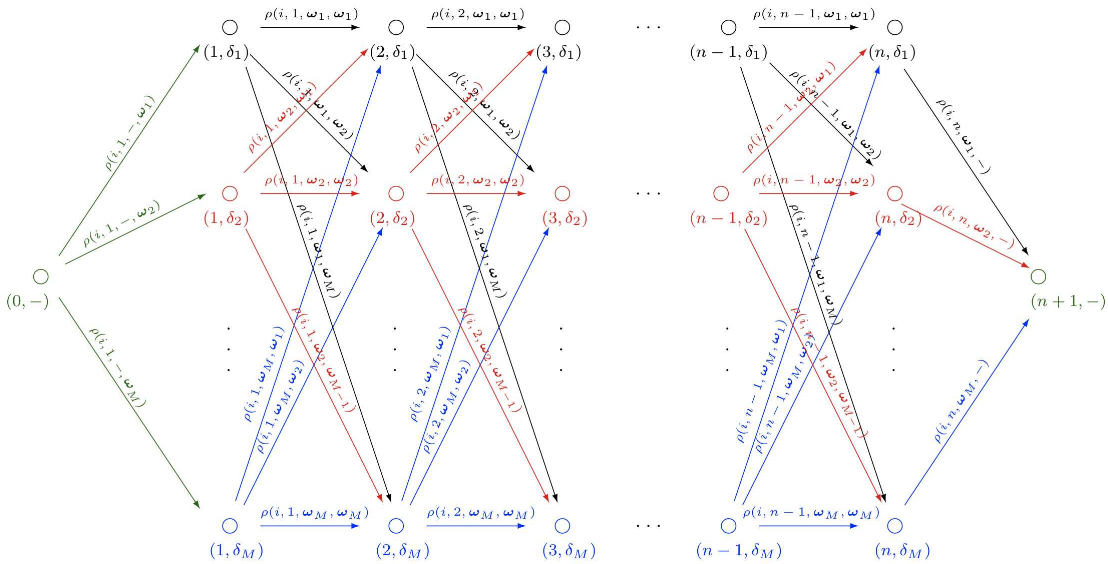


Fig. 7. Graph to minimize the local energy of a row considering M different possible displacements $\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_M$.

point. In the last step a GNC algorithm is used again in order to recover a sub–pixel accuracy. The experimental results, the first two synthetic and the latter two real, confirm the goodness of the technique presented in the paper.

REFERENCES

1. B. Horn and B. Schunk, Determining optical flow, *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.
2. M. Bertero, T. Poggio, and V. Torre, Ill-posed problems in early vision, *Proceedings of the IEEE*, vol. 76, pp. 869–889, 1988.
3. G. Aubert and P. Kornoprolist, *Mathematical Problems in Image Processing Applied Mathematical Sciences.* Springer Verlag, 2002.
4. E. Gamble and T. Poggio, Visual integration and detection of discontinuites: the key role of intesity edges, Tech. Rep. 970, MIT, 1987.
5. H. H. Nagel, On the estimation of optical flow: Relation between diferent approches and some new results, *Artificial Intelligence*, vol. 33, pp. 299–

324, 1987.

6. D. L. M.A. Arredondo, K. Lebart, Optical flow using textures, *Pattern Recognition Letters*, vol. 25, pp. 449–457, 2004.
7. H. H. Nagel and W. Enkelmann, An investigation of smoothness for the estimation of displacemnt vector fields from image sequences, *IEEE trans. Pattern Anal. Machine Intel.*, vol. 8, pp. 565–593, 1986.
8. E. Simoncelli, *Distrubuted Analyis and Representation of Visual Motion.* PhD thesis, MIT, 1993.
9. S. Raghavan, N. Gupta, and L. Kanal, Computing discontinuity–preserved image flow, in *Proceedings of ICPR92, the 11th International Conference on Pattern Recognition*, pp. 764–767, 1992.
10. J. Konrad and E. Dubois, Bayesian estimation of motion vector fields, *IEEE Trans. Pattern Anal. Machine Intel*, vol. 14, no. 9, pp. 367–383, 1992.
11. I. Gerace, A deterministic algorithm for optical flow estimation on directed graphs and the shortest path problem, tech. rep., Istituto di Elaborazione della Informazione, Pisa, Consiglio Nazionale delle Ricerche, 1995.
12. A. Blake and A. Zisserman, *Visual Recontruction.* Cambridge, Massachusetts: The MIT Press, 1987.
13. I. Gerace, Metodi di programmazione lineare per la ricostruzione visiva, tech. rep., Istituto di Elaborazione della Informazione, Pisa, Consiglio Nazionale delle Ricerche, 1995.
14. M. Dubuisson-Jolly and A. Gupta, Tracking deformable templates using a shortest path algorithm, *Computer Vision and Image Understanding*, vol. 81, pp. 26–45, 2001.
15. I. Gerace, Solving the sparse data image restoration problem by local minimization, in *Proceedings of IEEE, International Conference on Image Processing, ICIP*, 2005.
16. L. Alparone, M. Barni, F. Bartolini, and R. Caldelli, Regularization of optic flow estimates by means of weighted vector median filtering, *IEEE Trans. Image Processing*, vol. 8, no. 10, pp. 1462–1467, 1999.
17. G.Aubert, R. Deriche, and P.Kornprobst, Computing optical flow via variational techniques, *SIAM Appl. Math.,*, vol. 6, no. 1, pp. 156–182., 1999.
18. M. Nikolova, Markovian reconstruction using gnc approach, *IEEE Trans.Image Process.*, vol. 8, pp. 1204–1220, 1999.