Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational and Applied Mathematics 198 (2007) 396–413
JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS
www.elsevier.com/locate/cam

Image sequence inpainting: Towards numerical software for detection and removal of local missing data via motion estimation

L. D'Amore[a,*], L. Marcellino[a], A. Murli[a,b]

[a] *University of Naples, Federico II, Complesso Universitario M.S. Angelo, via Cintia, 80126 Naples, Italy*
[b] *Istitute of High Performance Computing and Networking, ICAR-CNR, Complesso Universitario M.S. Angelo, via Cintia, 80126 Naples, Italy*

Received 15 December 2004; received in revised form 15 September 2005

Abstract

Film restoration aims to remove or reduce various types of film and video defects in order to provide visual enhancements of image sequences. The automatic treatment of these defects is a challenge. Restoration is still performed by hand even if by using numerical techniques for retouching. This is a very intensive activity and great improvements, both in quality and in speed, can be obtained by using automatic or semiautomatic software.

This paper surveys the overall computational steps needed for the development of effective software tools to be actually used in a concrete application. In particular, here we focus on recovery and reconstruction of a particular *local random defect* of old black-and-white films, commonly referred to as "*blotch*". We start from the characterization of the degradation model both for detecting and for restoring the defect and deal with such inverse and ill-posed problem through edge preserving regularization. We employ a spatio-temporal interpolation for blotch removal where the initial approximation is given by interpolating along the motion trajectory data belonging to adjacent frames. Finally, we describe the numerical algorithm and some experimental results.

Keywords: Image sequence restoration; Motion estimate; Inverse problems; Edge preserving regularization; Multigrid

1. Introduction

Motion pictures represent an artistic and historic treasure. Unfortunately they are very fragile: the aging effects, bad maintenance, wear and tear are often cause of their deterioration. Movie restoration is still a very difficult problem, mainly because of the numerous types of degradation. Then, the first step is to try and classify them.

Table 1, introduced by the "*Commission Supérieure Technique de l'image et du son*" [9], characterizes the classic defects in terms of the amount of human interaction needed to detect and correct them.

	Interactive detection	Automatic detection
Interactive correction	Defect that can be confused with element of image	Emulsion tearing off or spots of any kind, non-stationary and wide
Automatic correction	Static defects of small areas, static spots. Color instability or sharpness differences	Defects that cannot be confused with image elements: non-stationary spots, flickering

* Corresponding author.
E-mail addresses: luisa.damore@dma.unina.it (L. D'Amore), livia.marcellino@dma.unina.it (L. Marcellino), almerico.murli@dma.unina.it (A. Murli).

0377-0427/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.09.030

Table 1
TreeD—comparison between the original blotch mask and the computed results

K	Number of detected pixels	DETECT (%)
Original mask	3009	100
10	4267	65
50	3273	89
100	2701	71
150	2035	79

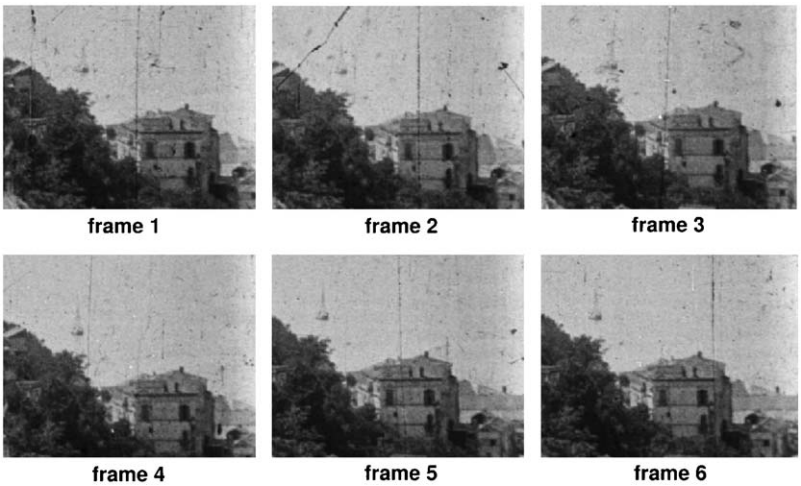

Fig. 1. Appearance of defects: blotches can have very different appearances.

Another type of classification can be made by means of spatio-temporal defect properties [16]. Usually, motion pictures suffer from four classes of defects, namely *global defects* (that appear on the entire image), *local defects* (restricted to a small part or a connected region of the image), *stationary defects* (which are static and do not depend on the time) and *random defects* (which are different from a frame to another).

The automatic treatment of these defects is a very difficult problem. Film restoration is actually performed by hand even using numerical techniques for retouching. Of course this is a very intensive and expensive activity and great improvements both in quality and in speed can be obtained by using automatic or semiautomatic software. This could reduce the costs and the efforts [8].

This paper aims to address computational difficulties arising in the production of numerical software for digital film restoration starting from the precise formulation of the degradation model, to the development of the algorithm, until to its implementation and testing with a quality assurance and quantitative validation of the final result. Thus, the main contribution is to describe and actually perform all steps needed towards the management of the application software, collecting and exploiting both numerical and mathematical tools, which have been sometimes already used separately in different contexts (machine vision, computer vision, etc.).

Here we focus on recovery and reconstruction of a particular *local random defect* of old black-and-white films, commonly referred to as "*blotch*".

When a foreign particle covers the film or the handling and aging causes loss of emulsion covering the film, bright or dark spots appear on the frame: these are the *blotches*. Blotches are the most common artefacts in old film sequences. They are caused by dirt or dust, holes, abrasions, water marks, hairs, spots. Moreover, it could happen that spots are present in only one frame ($\frac{1}{25}$ s), therefore they may appear and disappear rapidly (Fig. 1).

Many computational approaches for restoring missing data have been exploited for treating film blotches [6,12,14,18]. Difficulties related to blotch removal are due to the intrinsic ill-posedness which characterizes most image restoration problems. This means that knowledge of the degradation model is not enough to determine the desired solution.

Starting from [19] classical regularization functionals have been used imposing some a priori assumptions on the expected solution.

In our approach the underlying idea for *detecting and removing* such kind of defects is to use information given by the movement in the sequence, which can be perceived through the displacements of brightness patterns over time. We employ a spatio-temporal interpolation for blotch removal where the initial approximation is given interpolating along the motion trajectory data belonging to adjacent frames.

The paper is organized as follows: Section 2 introduces the blotch degradation model, definitions of motion field and motion trajectory. In Section 3, we describe the computation of the blotch domain, while in Section 4 the edge-preserving regularization we use for blotch removal and for the motion estimate is reviewed. In Section 5, numerical approach and its related algorithm are described. Finally, in Section 6, validation of the experimental results in terms of the reliability of the solution together with the automatic stopping criterion is presented, while in Section 7, conclusions and remarks about the achieved results are discussed.

2. The degradation model

Following [12], one of the properties which characterize blotches is that: "*the intensity inside a blotch is significantly different from its neighboring uncorrupted intensities*". This degradation is not limited to a single pixel but can occur as variable size regions highly contrasted with the surrounding pixels (Fig. 2).

Such assumption allows to formally characterize both the blotch domain and the intensity inside the blotch region. To this aim, let us first recall some basic definitions used in the following, and introduce *the degradation model* that we are going to use.

Definition 1. Let $J \subset \Re$ be the time interval, and let the brightness of a sequence image be $I : t \in J \longrightarrow P(t) \in \Omega \longrightarrow I(P(t), t) \in \Re$, where $\Omega = \Omega_x \times \Omega_y \subset \Re^2$ is the image plane. Given $t \in J$, we denote by $\widetilde{I}(P(t), t)$ a noisy sequence, where [13]

$$\widetilde{I}(P(t), t) = [1 - b(P(t))] \cdot I(P(t), t) + b(P(t)) \cdot I^B(P(t), t) \tag{1}$$

and

- b is the blotch characteristic function, which defines the blotch domain $B \subset \Omega$:

$$b : P(t) \in \Omega \longrightarrow b(P(t)) = \begin{cases} 1 & \text{if } P(t) \in B, \\ 0 & \text{if } P(t) \in \Omega \backslash B, \end{cases}$$

- I^B refers to the brightness level at B.

The blotch degradation model (1) leads to the following *inverse and ill-posed problem*:

Fig. 2. On the left, only one simple blotch: a spot. On the right, a more complex case: many different small blotches.

Definition 2. Given $I(P(t), t), \forall t \in J, \forall P(t) \in \Omega\backslash B$, *determine the characteristic function b* (*blotch detection*), and then *compute the missing data on B* (*blotch removal*).

It is worth noting that, according to the digital film restoration process where each defect is treated separately from the others, the blotch degradation model (1) assumes that the function $I(P(t), t)$, $P(t) \in \Omega\backslash B$ is noise free. Moreover, we assume that the intensity inside the blotch is completely unknown, this is why we deal with missing data inside the degraded region.

3. Blotch detection

Definition 3. At each point $P(t) \in \Omega$, the motion trajectory (Fig. 3) is the line or arc L, defined by successive positions of $P(t)$, as $t \in J$. The equation for L is

$$L : \begin{cases} \Delta x(P(t)) = \Delta t\, v_x(P(t)), \\ \Delta y(P(t)) = \Delta t\, v_y(P(t)). \end{cases}$$

Let $t - \Delta t, t, t + \Delta t \in J$. The positive and the negative directions of the motion trajectory are defined by

$$L^+ : \begin{cases} \Delta x^+ = \Delta t^+ \cdot v_x^+(P(t - \Delta t)), \\ \Delta y^+ = \Delta t^+ \cdot v_y^+(P(t - \Delta t)), \end{cases} \quad L^- : \begin{cases} \Delta x^- = \Delta t^- \cdot v_x^-(P(t + \Delta t)), \\ \Delta y^- = \Delta t^- \cdot v_y^-(P(t + \Delta t)), \end{cases}$$

where

$$v^+(P(t - \Delta t)) = (v_x^+(P(t - \Delta t)),\ v_y^+(P(t - \Delta t))),$$

$$v^-(P(t + \Delta t)) = (v_x^-(P(t + \Delta t)),\ v_y^-(P(t + \Delta t)))$$

are, respectively, the motion field, at each point $P(t - \Delta t) \in \Omega$, from $t - \Delta t$ to t; and the *motion field*, at each point $P(t + \Delta t) \in \Omega$, from $t + \Delta t$ to t.

Definition 4. The directional derivatives of I along the *positive direction* and *negative direction* of the motion trajectory are

$$\frac{\partial I}{\partial L^+} = \lim_{\Delta t^+ \to 0} \frac{I(x(t - \Delta t) + \Delta x^+, y(t - \Delta t) + \Delta y^+, t - \Delta t) - I(x(t), y(t), t)}{\Delta t^+},$$

$$\frac{\partial I}{\partial L^-} = \lim_{\Delta t^- \to 0} \frac{I(x(t + \Delta t) - \Delta x^-, y(t + \Delta t) - \Delta y^-, t + \Delta t) - I(x(t), y(t), t)}{\Delta t^-}.$$

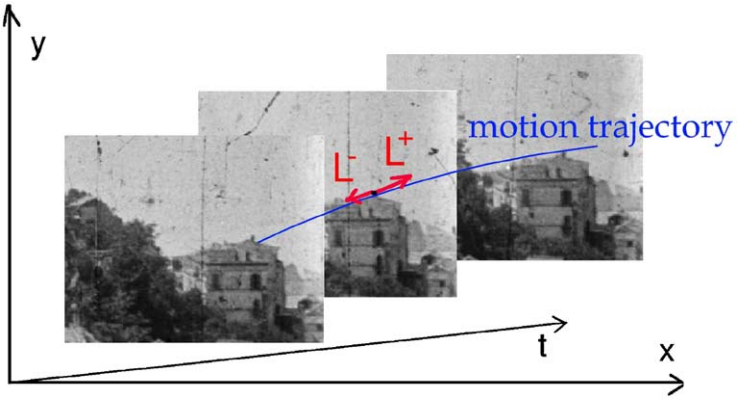

Fig. 3. The arc of line L, which defines the motion trajectory.

In order to numerically compute the blotch domain B we rely on the relationship between the *motion field* of a given sequence and the *brightness level* of each image of the sequence [2,7]. More precisely, we use the assumption that "*sequence brightness level is constant along the motion trajectory*" [11].

This means that directional derivatives of I computed along positive direction L^+ and negative direction L^- of the motion trajectory are zero, i.e. for each $P(t) \in L$

$$\frac{\partial}{\partial L^+} I(P(t), t) = 0, \quad \frac{\partial}{\partial L^-} I(P(t), t) = 0. \tag{2}$$

Then at each $t \in J$, the blotch domain and its characteristic function can be determined by

$$B = \left\{ P(t) \in \Omega : \min \left\{ \left| \frac{\partial \widetilde{I}}{\partial L^+} \right|, \left| \frac{\partial \widetilde{I}}{\partial L^-} \right| \right\} \neq 0 \right\}$$

and

$$b(P(t)) = 0 \Rightarrow \widetilde{I}(P(t), t) = I(P(t), t), \quad P(t) \in \Omega \backslash B,$$

$$b(P(t)) = 1 \Rightarrow \widetilde{I}(P(t), t) = I^B(P(t), t), \quad P(t) \in B \subset \Omega.$$

Once the function b has been computed, the recovery of the unknown intensity on B concerns the computation of I^B at spatial locations belonging to B. The next section describes the regularization approach that we are going to use.

4. Blotch removal

From now on we assume that $I \in L^2(\Omega)$. The starting point of a blotch removal algorithm is to compute I solving a *least square* problem

$$\min \|I^0 - I\|,$$

where $I^0 \in L^2(\Omega)$ is some given initial data, and $\|\cdot\|$ is the standard norm in L^2. As already said, additional constraints are needed to handle the ill-posedness of this inverse problem.

In order to preserve discontinuities of I we consider the regularized problem:

$$\min_{I \in W^{1,2}(\Omega)} E(I) = \min_{I \in W^{1,2}(\Omega)} \{\|I^0 - I\|_{L^2} + \lambda \|\phi(\|\nabla I\|_{L^2})\|_{L^1}\}, \tag{3}$$

where the function ϕ smooths at locations where the variation of intensity is small (low gradients) and preserves discontinuities in neighborhood of high gradients.

Consider the *dual functional* [10]

$$E^{\mathrm{d}}(I, b) = \int_\Omega |I - I^0|^2 \, \mathrm{d}\underline{x} + \lambda \int_\Omega (b\|\nabla I\|_{L^2}^2 + \psi(b)) \, \mathrm{d}\underline{x}, \quad \underline{x} = (x, y), \tag{4}$$

which is quadratic with respect to I, once b is fixed and, by the same way it is quadratic with respect to b, once I is fixed. The function ψ must be chosen according to the ϕ function, in agreement with the *half-quadratic regularization* [5]. Let

$$\begin{cases} \widehat{b} = \arg \min_{b \in L^2(\Omega)} E^{\mathrm{d}}(\cdot, b), \\ I_{\widehat{b}} = \arg \min_{I \in W^{1,2}(\Omega)} E^{\mathrm{d}}(I, \cdot), \end{cases}$$

where $E^{\mathrm{d}}(\cdot, b)$ means that the first argument I has been fixed and $E^{\mathrm{d}}(I, \cdot)$ means that the second argument b has been fixed.

By means of alternate minimization over b and I_b, the best fits $\widehat{b}$ and $I_{\widehat{b}}^B$ can be approximated using the *alternate fixed point scheme* where the initial data are I^0 and $b^0 = \phi'(\nabla I^0)/2\|\nabla I^0\|$.

At each step $n = 0, 1, 2, \ldots$ the iterates b^{n+1} can be expressed as

$$b^{n+1} = \frac{\phi'(\nabla I^n)}{2\|\nabla I^n\|}, \tag{5}$$

and similarly, at each step $n = 0, 1, 2, \ldots$ the iterates I^{n+1} satisfy the *Euler–Lagrange* equation

$$I^{n+1} = G \star I^n - \lambda \operatorname{div}(b^{n+1}\nabla I^n), \tag{6}$$

where the iterates I^n, $n = 0, 1, 2, \ldots$, have been convolved with a gaussian function G [17].

The underlying idea of the spatio-temporal interpolation that we are going to use is the following assumption which characterizes the presence of blotches in a video sequence: "*the blotches are independent with respect to the time interval then they hardly ever occur at the same spatial location in successive frames*" [13].

This means that the degraded brightness can be recovered by using information on the neighboring frames and interpolating along motion trajectory.

Following the point $P(t-\Delta t)$ along the *positive* motion trajectory, we compute its position at time t using the motion displacements Δx^+ and Δy^+:

$$\begin{cases} x(t) = x(t-\Delta t) + \Delta x^+, \\ y(t) = y(t-\Delta t) + \Delta y^+, \end{cases}$$

then let

$$I^C_{t-\Delta t}(x(t), y(t), t) = I(x(t-\Delta t) + \Delta x^+, y(t-\Delta t) + \Delta y^+, t-\Delta t). \tag{7}$$

In the same way, following the point $P(t+\Delta t)$ along the *negative* motion trajectory, we compute its position at time t, using the motion displacements Δx^- and Δy^-:

$$\begin{cases} x(t) = x(t+\Delta t) - \Delta x^-, \\ y(t) = y(t+\Delta t) - \Delta y^-, \end{cases}$$

then let

$$I^C_{t+\Delta t}(x(t), y(t), t) = I(x(t+\Delta t) - \Delta x^-, y(t+\Delta t) - \Delta y^-, t+\Delta t). \tag{8}$$

We assume as reference data in the regularization functional (3) and (4), the following:

$$I^0(x(t), y(t), t) = \frac{I^C_{t-\Delta t}(x(t), y(t), t) + I^C_{t+\Delta t}(x(t), y(t), t)}{2}. \tag{9}$$

Blotch removal algorithm

for $(x, y) \in B$ ***do***
  let be I^0 as in (9)
  for $n = 1, 2, \ldots$ ***repeat***
  $b^{n+1} = \dfrac{\phi'(\nabla I^n)}{2\|\nabla I^n\|}$
  $I^{n+1} = G \star I^n - \lambda \operatorname{div}(b^{n+1}\nabla I^n)$
  until (convergence)
endfor

Iterations are terminated as soon as the *stopping criterion*, based on the *mean square error* (MSE)

$$\mathrm{MSE}(n+1) = \frac{1}{N_B \times M_B} \sum_{x=1}^{N_B} \sum_{y=1}^{M_B} [I^{n+1}(x, y) - I^n(x, y)]^2,$$

where $N_B \times M_B$ is the cardinality of B, is satisfied.

Finally, observe that we considered the classical approach for solving inverse and ill-posed problems through minimization of regularization functional. An equivalent approach is to perform anisotropic diffusion to obtain an enhanced image dealing directly with the PDE equation which describes the diffusion filter [1,4,13,15]. These two approaches sometimes can be related. Indeed, relation (6) can be seen as a forward explicit scheme arising from a suitable diffusion filter whose coefficient depends on the edge preserving function (see [20] for details).

In order to compute $\Delta x^{\pm}$ and $\Delta y^{\pm}$ in (7) and (8) we need to estimate the *motion field*. The next section describes approximation of motion field through the computation of the optical flow.

4.1. Motion field

As soon as we consider a sequence there is the idea of motion. Unfortunately, we are not able to measure the motion field (the projection on the image plane of the 3D velocity), but what we are able to perceive is just an apparent motion, also called *optical flow*. This apparent motion is observable through intensity variations.

Different methods have been proposed to compute the optical flow. These can be essentially classified as *matching* techniques, which compare two given images pixel by pixel in terms of the brightness in order to determine the motion vectors; and *variational* methods (also called *regularization techniques*) [2].

The starting point of variational methods is the *optical flow constraint equation*:

$$\frac{\mathrm{d}}{\mathrm{d}t} I(P(t), t) = \frac{\mathrm{d}}{\mathrm{d}t} I(x(t), y(t), t) = 0 \Longleftrightarrow \overrightarrow{v} \times \nabla I + I_t = 0,$$

where

$$\nabla I = \left(\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right), \quad I_t = \frac{\partial I}{\partial t}, \quad \overrightarrow{v} = (v_x, v_y) = \left(\frac{\mathrm{d}}{\mathrm{d}t} x(t), \frac{\mathrm{d}}{\mathrm{d}t} y(t) \right).$$

This equation refers to the assumption that "*the intensity of a point keeps constant along its trajectory*" [11].

The optical flow equation is again an ill-posed problem and additional constraints are required. Starting from the seminal work of Horn and Schunck [11], which proposed a quadratic constraint that cannot take into account the discontinuities of the flow field, different regularizations have been proposed [1,2,11]. Usually, the additional term formalizes the assumption that "*near points have the same velocity vector*". Thus, the flow field varies smoothly in the space and the spatial gradients should be close to zero.

Following [1], we employ the regularization functional

$$\begin{aligned} &\min_{\overrightarrow{v}=(v_x,v_y)\in W^{1,2}(\Omega)} E(v_x, v_y) \\ &\quad = \min_{\overrightarrow{v}\in W^{1,2}(\Omega)} \{ \| \nabla I \times \overrightarrow{v} + I_t \|_{L^2} + \alpha_r \phi(\| \nabla v_x \|_{L^2}) + \phi(\| \nabla v_y \|_{L^2}) \}, \end{aligned} \tag{10}$$

where ϕ is such that at locations where the variations of v_x and v_y are small smoothing is encouraged, and in neighborhood of edges discontinuity of the gradient must be preserved.

We again consider the *dual functional* and the corresponding *dual function* ψ [10]:

$$\begin{aligned} E^{\mathrm{d}}(\overrightarrow{v}, a, b_x, b_y) = &\int_{\Omega} (a|\overrightarrow{v} \times \nabla I + I_t|^2) \, \mathrm{d}\underline{x} \\ &+ \int_{\Omega} [(b_x \|\nabla v_x\|^2 + \psi(b_x)) + (b_y \|\nabla v_y\|^2 + \psi(b_y))] \, \mathrm{d}\underline{x} \end{aligned}$$

that we first minimize with respect to $(\cdot, \cdot, b_x, b_y)$, then with respect to $(\cdot, a, \cdot, \cdot)$ and then with respect to $(\overrightarrow{v}, \cdot, \cdot, \cdot)$, by using again the *fixed point scheme*.

Let $\overrightarrow{v}^0 = (v_x^0, v_y^0) = (0, 0)$:

$$\begin{cases} (b_x^{n+1}, b_y^{n+1}) = \arg \min_{(b_x,b_y)} E^{\mathrm{d}}(v_x^n, v_y^n, a^n, b_x, b_y), \\ a^{n+1} = \arg \min_{a} E^{\mathrm{d}}(v_x^n, v_y^n, a, b_x^{n+1} b_y^{n+1}), \\ (v_x^{n+1}, v_y^{n+1}) = \arg \min_{\overrightarrow{v}=(v_x,v_y)} E^{\mathrm{d}}(v_x, v_y, a^{n+1}, b_x^{n+1}, b_y^{n+1}). \end{cases}$$

For each $n = 0, 1, 2, \ldots$, the minimum with respect to parameters (b_x, b_y) is reached at

$$b_x^{n+1} = \frac{\phi'(\|\nabla v_x^n\|)}{2\|\nabla v_x^n\|}, \quad b_y^{n+1} = \frac{\phi'(\|\nabla v_y^n\|)}{2\|\nabla v_y^n\|}, \tag{11}$$

where $\|\cdot\|$ denotes the standard euclidean norm in $\Re^2$. Similarly, the minimum with respect to parameter a^{n+1}, at step $n+1$, is reached at

$$a^{n+1} = \frac{1}{2|\overrightarrow{v}^n \times \nabla I + I_t|}, \tag{12}$$

where $|\cdot|$ denotes the absolute value of a real number. Finally, the minimum with respect v_x, v_y, at step $n+1$, satisfies the Euler–Lagrange equations

$$\begin{cases} a^{n+1}(\overrightarrow{v}^n \times \nabla I + I_t)I_x + \alpha^r \operatorname{div}(b_x^n \nabla v_x^n) + c(\underline{x})[v_x^{n+1} - v_x^n] = 0, \\ a^{n+1}(\overrightarrow{v}^n \times \nabla I + I_t)I_y + \alpha^r \operatorname{div}(b_y^n \nabla v_y^n) + c(\underline{x})[v_y^{n+1} - v_y^n] = 0, \end{cases} \tag{13}$$

where α^r and α^h are positive constants and function c is defined as follows:

$$c(x) = \begin{cases} 0 & \text{if } \dfrac{\|\nabla I\|}{\max(\|\nabla I\|)} \geqslant S_g \text{ and } \dfrac{|I_t|}{\max(|I_t|)} \geqslant S_t, \\ 1 & \text{otherwise,} \end{cases}$$

S_g and S_t being thresholds related to the spatial and temporal derivatives, respectively. Relations (11)–(13) give rise to the explicit scheme for optical flow computation.

Optical flow algorithm

for $(x, y) \in \Omega$ ***do***
$v_x^0 = 0 \quad v_y^0 = 0$
for $n = 1, 2, \ldots$ ***repeat***
$$a^{n+1} = \frac{1}{2|\overrightarrow{v}^n \times \nabla I + I_t|} \quad b_x^{n+1} = \frac{\phi_2'(\|\nabla v_x^n\|)}{2\|\nabla v_x^n\|}, \quad b_y^{n+1} = \frac{\phi_2'(\|\nabla v_y^n\|)}{2\|\nabla v_y^n\|}$$
$$\alpha^h c(\underline{x}) v_x^{n+1} = c(\underline{x}) v_x^n + a^{n+1}(\overrightarrow{v}^n \times \nabla I + I_t)I_x + \alpha^r \operatorname{div}(b_x^{n+1} \nabla v_x^n)$$
$$\alpha^h c(\underline{x}) v_y^{n+1} = c(\underline{x}) v_x^n + a^{n+1}(\overrightarrow{v}^n \times \nabla I + I_t)I_y + \alpha^r \operatorname{div}(b_y^{n+1} \nabla v_y^n)$$
until (convergence)
endfor

Iterations are terminated as soon as the *stopping criterion*, based on the *mean square error* (MSE):

$$\mathrm{MSE}(n+1) = \frac{1}{N \times M} \sum_{x=1}^{M} \sum_{y=1}^{N} [|\overrightarrow{v}^{n+1}(x, y)| - |\overrightarrow{v}^n(x, y)|]^2,$$

where $N \times M$ is the size of Ω, and $|\overrightarrow{v}(x, y)| = \sqrt{v_x^2(x, y) + v_y^2(x, y)}$ is satisfied.

5. Numerical approximation

Let $(v_x^0, v_y^0) = (0, 0)$ and, following [1], we consider $\phi(s) = 2\sqrt{1 + s^2} - 2$ with the corresponding $\psi(s) = s + 1/s$. Here we describe the discretization schemes we considered both for the optical flow computation and for the blotch removal algorithms.

For spatial discretizations of $\Delta^y I(x, y, t)$ and $\Delta^y I(x, y, t)$ we use central finite differences on ghost points $(x + \Delta x/2, y + \Delta y/2)$:

$$\frac{\partial}{\partial x} I(x, y, t) \approx \Delta^x I(x, y, t) = I\left(x + \frac{\Delta x}{2}, y, t\right) - I(x, y, t),$$

$$\frac{\partial}{\partial y} I(x, y, t) \approx \Delta^y I(x, y, t) = I\left(x, y + \frac{\Delta y}{2}, t\right) - I(x, y, t)$$

and

$$\frac{\partial}{\partial x} v_x^n(x, y, t) \approx \Delta^x v_x^n = v_x^n\left(x + \frac{\Delta x}{2}, y, t\right) - v_x^n(x, y, t),$$

$$\frac{\partial}{\partial y} v_x^n(x, y, t) \approx \Delta^y v_x^n = v_x^n\left(x, y + \frac{\Delta y}{2}, t\right) - v_x^n(x, y, t),$$

$$\frac{\partial}{\partial x} v_y^n(x, y, t) \approx \Delta^x v_y^n = v_y^n\left(x + \frac{\Delta x}{2}, y, t\right) - v_y^n(x, y, t),$$

$$\frac{\partial}{\partial y} v_y^n(x, y, t) \approx \Delta^y v_y^n = v_y^n\left(x, y + \frac{\Delta y}{2}, t\right) - v_y^n(x, y, t).$$

Let us consider the terms

- $\operatorname{div}(b^{n+1} \cdot \nabla I^n)$ in (6),
- $\operatorname{div}(b_x^{n+1} \cdot \nabla v_x^n)$, $\operatorname{div}(b_y^{n+1} \cdot \nabla v_y^n)$ in (13).

If we indicate by d^{n+1} the b_x^{n+1}, b_y^{n+1}, and b^{n+1}, while a^n stands for ∇I^n it is

$$\begin{aligned}\operatorname{div}(d^{n+1} \cdot \nabla a^n) &= \frac{\partial}{\partial x}\left(d^{n+1} \cdot \frac{\partial}{\partial x} a^n\right) + \frac{\partial}{\partial y}\left(d^{n+1} \cdot \frac{\partial}{\partial y} a^n\right) \\ &\approx \Delta^x(d^{n+1} \cdot \Delta^x a^n) + \Delta^y(d^{n+1} \cdot \Delta^y a^n).\end{aligned}$$

Moreover, consider at (x, y) the matrices P and D:

$$P = \begin{bmatrix} 0 & d^{n+1}\left(x - \frac{\Delta x}{2}, y\right) & 0 \\ d^{n+1}\left(x, y - \frac{\Delta y}{2}\right) & -S^P & d^{n+1}\left(x, y + \frac{\Delta y}{2}\right) \\ 0 & d^{n+1}\left(x + \frac{\Delta x}{2}, y\right) & 0 \end{bmatrix},$$

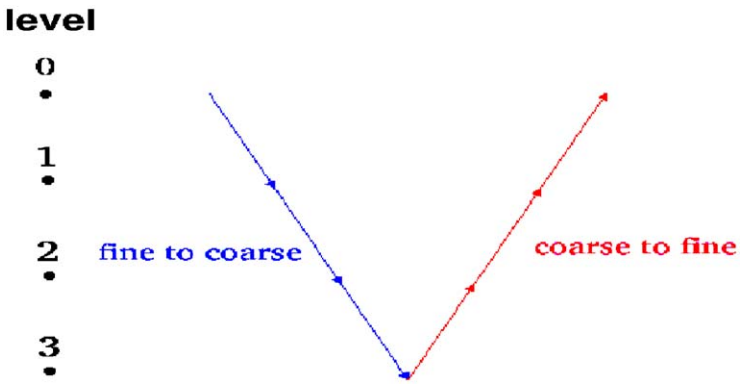

Fig. 4. V-cycle multigrid.

where

$$S^P = d^{n+1}\left(x - \frac{\Delta x}{2}, y\right) + d^{n+1}\left(x + \frac{\Delta x}{2}, y\right) + d^{n+1}\left(x, y - \frac{\Delta y}{2}\right) + d^{n+1}\left(x, y + \frac{\Delta y}{2}\right)$$

and

$$D = \begin{bmatrix} d^{n+1}\left(x - \frac{\Delta x}{2}, y - \frac{1}{2}\right) & 0 & d^{n+1}\left(x - \frac{\Delta x}{2}, y - \frac{\Delta y}{2}\right) \\ 0 & -S^D & 0 \\ d^{n+1}\left(x + \frac{\Delta x}{2}, y + \frac{\Delta y}{2}\right) & 0 & d^{n+1}\left(x + \frac{\Delta x}{2}, y + \frac{\Delta y}{2}\right) \end{bmatrix},$$

where

$$S^D = d^{n+1}\left(x - \frac{\Delta x}{2}, y - \frac{\Delta y}{2}\right) + d^{n+1}\left(x - \frac{\Delta x}{2}, y + \frac{\Delta y}{2}\right) d^{n+1} + d^{n+1}\left(x + \frac{\Delta x}{2}, y - \frac{\Delta y}{2}\right) + d^{n+1}\left(x + \frac{\Delta x}{2}, y + \frac{\Delta y}{2}\right).$$

Then it is

$$\operatorname{div}(d^{n+1}\nabla a^n)(x, y) = \alpha_P P \star a^n(x, y) + \alpha_D D \star a^n(x, y),$$

where α_P, α_D are such that $\alpha_P + 2\alpha_D = 1$. Two choices have been considered [1]:

- (α_P, α_D) are fixed (for instance equal to $(\frac{1}{2}, \frac{1}{4})$), which means that a preference to main direction is given;
- (α_P, α_D) are chosen according to the orientation of the gradient of a.

Observe that the intensity displacements computed by 2-stencil forward finite differences depend on h, that is, the grid mesh size. In order to take into account both small and large displacements and then to improve the accuracy of the computed optical flow a *multigrid approach* is usually employed [3].

Multigrid methods for optical flow computation have been exploited in the last years because these allow to compute accurate results much faster than not hierarchical solvers [21]. Here we use the simplest *V-cycle cascading multigrid* (Fig. 4). Observe that in case of digital film sequences, at least 25 frames per second are recorded, then multiresolution actually improves the computed result if $v > 25\,\mathrm{h}$ where v indicates the motion field measured in seconds.

The multigrid approach is essentially based on a hierarchical approximation made of two main steps: *decomposition* and *multigrid correction*.

We first construct the multiresolution pyramid moving from the given image mesh that represents the finest grid towards the coarsest resolution level (*fine-to-coarse decomposition*). Then, starting from the coarsest level we compute

the optical flow moving towards the finest, upgrading the computed solution (*coarse-to-fine correction*), as detailed in the following:

Fine-to-coarse decomposition: In the first step the given image is reproduced on a set of different levels of resolution, which correspond to the approximation subspaces. Starting from the given image I, whose domain Ω is made of $N \times M$ pixels, where $N = 2^{2n}$ and $M = 2^{2m}$ and stepsize h, we construct the multiresolution pyramid using *orthonormal gaussian wavelets*:

$$\omega(x, y) = \frac{1}{4} \cdot \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}.$$

The multiresolution pyramid is made of a *sequence* of images: $\{\Omega_i : \Omega_0 \equiv \Omega\}_{0 \leqslant i \leqslant \text{level}}$ where, at each level i, $1 \leqslant i \leqslant$ level:

- Ω_i is obtained from the previous one Ω_{i-1}, doubling the step size $h_i = 2h_{i-1}$;
- the brightness I_i, at each point $(x, y, t) \in \Omega_i \times J$, is computed by

$$I_i(x, y, t) = I_{i-1}(x, y, t) \star \omega(x, y),$$

where $\star$ is the convolution operator.

The parameter *level* depends on the image size: at each step i, if $N/2^i \geqslant \min N$ and $M/2^i \geqslant \min M$ (where $\min N$ and $\min M$ are two given thresholds), then the decomposition is performed.

Coarse-to-fine correction: In the second step, the problem is solved on each subspace proceeding from the coarsest towards the finest levels, according to a multigrid correction with a projection and a smoother at each step; the smoothing operator is the solution of the Euler–Lagrange equation on each level, while projection is made by duplication.

The vector field is computed at each level starting from the coarsest and moving towards the next finer grid. More precisely, the optical flow is first computed at level i, then it is projected onto the next finer level $i - 1$ where it is added to the local approximation.

Multigrid optical flow computation

---

for $(x, y, t) \in \Omega \times J$ ***do***
for $(i = level, 0, step\text{-}1)$ ***do***
%% extension by duplication:
if $(i = level)$ ***then***
$\delta \vec{v}_i(x, y, t) := 0$
else
$\delta \vec{v}_i(x, y, t) := \vec{v}_{i+1}\left(\frac{x}{2}, \frac{y}{2}, t\right)$
endif
%% computation of $\vec{v}_i$ *by using the*
%% optical flow algorithm (see page 12)
%% update
$\vec{v}_i(x, y, t) = \vec{v}_i(x, y, t) + \delta \vec{v}_i(x, y, t)$
endfor
endfor

Now, we describe the algorithm for the blotch detection and removal. Starting from the following quantities:

$$I^C_{t-\Delta t}(x(t), y(t), t) = I(x(t - \Delta t) + \Delta x^+, y(t - \Delta t) + \Delta y^+, t - \Delta t),$$

$$I^C_{t+\Delta t}(x(t), y(t), t) = I(x(t + \Delta t) - \Delta x^-, y(t + \Delta t) - \Delta y^-, t + \Delta t),$$

once t has been fixed, we consider

$$W_{t-\Delta t}(x, y) = |I^C_{t-\Delta t}(x, y, t) - I(x, y, t)|, \quad W_{t+\Delta t}(x, y) = |I^C_{t+\Delta t}(x, y, t) - I(x, y, t)|,$$

which are used to compute the *blotch mask*, which gives the characteristic function b:

$$B(x, y) = \begin{cases} 1 & \text{if } \min\{W_{t-\Delta t}(x, y), W_{t+\Delta t}(x, y)\} \geqslant K, \\ 0 & \text{otherwise,} \end{cases}$$

where K is a given tolerance. This mask is the reference binary image $B(x, y)$ at time t, where the pixels corresponding to the blotch domain appear.

Blotch removal algorithm

for $(x, y) \in B$ ***do***
$I^0 = \dfrac{I^C_{t-\Delta t} + I^C_{t+\Delta t}}{2}, \quad b^0 = \dfrac{\phi'(\nabla I^0)}{2\|\nabla I^0\|}$
for $n = 1, 2, \ldots$ ***repeat***
$b^{n+1} = \dfrac{\phi'(\nabla I^n)}{2\|\nabla I^n\|}$
$I^{n+1} = G \star I^n - \lambda \operatorname{div}(b^{n+1} \nabla I^n)$
until (convergence)
endfor

It is $\phi(s) = 2\sqrt{1 + s^2} - 2$, then $b^{n+1} = \phi'(\|\nabla I^n\|)/2\|\nabla I^n\| = 1/\sqrt{1 + \|\nabla I^n\|^2}$.
Moreover, regarding the gradient $\nabla I^n = ((\partial/\partial x) I^n, (\partial/\partial y) I^n)$, it is

$$\frac{\partial}{\partial x} I^n(x, y, t) \approx \Delta^x I^n(x, y, t) = I^n\left(x + \frac{1}{2}, y, t\right) - I^n(x, y, t),$$

$$\frac{\partial}{\partial y} I^n(x, y, t) \approx \Delta^y I^n(x, y, t) = I^n\left(x, y + \frac{1}{2}, t\right) - I^n(x, y, t).$$

Therefore, at step $n + 1$ the function I^{n+1} is computed as follows:

$$I^n(x, y, t) = \left[\sum_{h=-1}^{1} \sum_{k=-1}^{1} G(h, k) I^n(x + h, y + k, t)\right] + \lambda[\operatorname{div}(b^{n+1} \cdot \nabla I^n)],$$

where $G(x, y)$ is the discrete approximation of the two-dimensional gaussian function, and $\sigma = 1.0$, while the divergence term is discretized using the finite difference scheme given previously.

6. Results and observations

We carried out some experiments aimed to verify both the accuracy of the computed results and the robustness of the algorithm. To this aim we consider real and synthetic sequences. Here we show only two tests: the first one concerns a simulated sequence that we denote by *TreeD*, where the motion is due to a zoom of the camera; the second one refers to a sequence extracted from an old movie about *Naples*.

All the experiments have been performed using the of IBM RISC 6000.

In order to validate the computed results we use the following quality measures:

- MSE: Measure of the average relative error on the computed solution.
- DETECT: Percentage of detected pixels in the right position computed by the detection algorithm.
- Cross-section profile: The bidimensional function that describes the gray level intensity versus the x-axis at a given y, of the original image and of the restored one. The cross-section profile allows to exactly compare pixel by pixel the restored brightness intensity with the true one.

- The average angular error:

$$\text{average}_{\text{err}} = \left(\sum \psi_E(x, y)/\text{np}\right),$$

where

$$\psi_E(x, y) = \text{arcosen}(v_c(x, y) \times v_e(x, y)),$$

$v_c(x, y)$ is the computed vector field, $v_e(x, y)$ is the exact vector field, np is the number of pixels. The symbol $\times$ denotes the scalar product operator.

Fig. 5. Sequence *TreeD*, image size: 150 × 150 pixels.

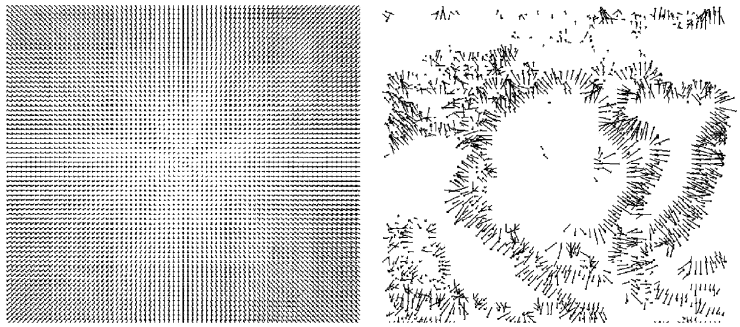

Fig. 6. The original optical flow (left) and the computed (right) optical flow. Sg $= 0.1$, St $= 0.1$. Cpu: 6.67 s.

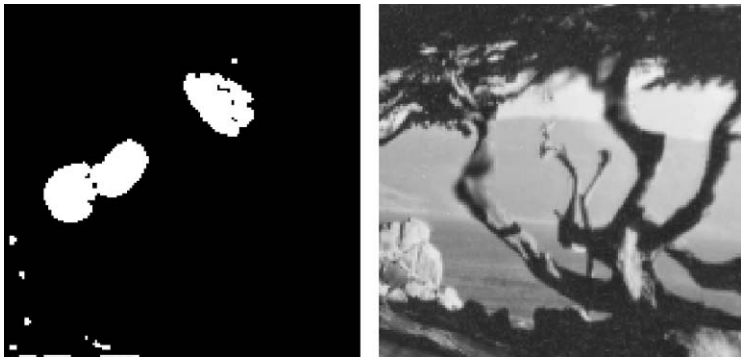

Fig. 7. Sequence *TreeD*. On the left the computed blotch mask. On the right the restored frame. Cpu: 1.645 s. $K = 50$.

- The standard deviation:

$$\mathrm{StDev} = \sqrt{\left(\left|\left(\sum \psi_E(x,y)\right)^2 - \mathrm{np} \times \mathrm{average}_{\mathrm{err}}^2\right)\right/ (\mathrm{np}-1)}.$$

Both the average angular error and the standard deviation are used for measuring the accuracy of the computed vector field.

Regarding the first test, i.e. the sequence that we denote by *Sequence TreeD*, three frames are shown in Fig. 5. Observe the two large black blotches corresponding to the degraded regions.

Results are shown in Figs. 6 and 7. Note that the true motion field, on the left, clearly appears to be directed towards the center of the image indicating the zoom of the camera. On the right, the computed optical flow field is shown, which is more evident along the contours of the objects, as expected, because it measures the brightness variations.

In Fig. 15 the MSE plots (in *logarithmic scale*) are shown. We set the tolerance equals to $\mathrm{Tol} = 10^{-3}$ both for the optical flow computation stopping criterion and for the blotch removal. This means that the iterative algorithms proceed until $\mathrm{MSE}(n) \leqslant \mathrm{Tol}$. The required accuracy on the computed optical flow after about 15 iterations is within three decimal digits. In the same way, the computed solution after about 10 iterations inside the blotch region is within 3 decimal digits, according to the numerical representation of the brightness intensity in [0, 255].

In Table 1 we report the number of pixels belonging to the original blotch and, for each choice of $K \in [0, 255]$, the parameter DETECT. and 9

Observe that if K is too small (i.e. $K < 50$) the number of detected pixels grows, but these may not belong all to the blotch domain. Instead, increasing K beyond 100 the number of detected pixels decreases, sometimes together with the corrected locations. We find that the optimal value of K is about 50–60, because it gives the right trade off with respect to true and false positions.

Figs. 8 and 9 show the computed intensity function in terms of the *cross-section profile*. Inside the blotch domain, the box line (zero value) in the center of the blotch clearly appears, while the other two lines that are quite near refer to the original (circle line) and to the restored brightness (star line). In Fig. 10 the MSE plots of the optical flow and of the blotch removal alogorithms are shown.

Finally, Table 2 summarizes the average error and the standard deviation corresponding to the optical flow field of the sequence *TreeD*. Results are compared with the most common regularization functionals reported in [2].

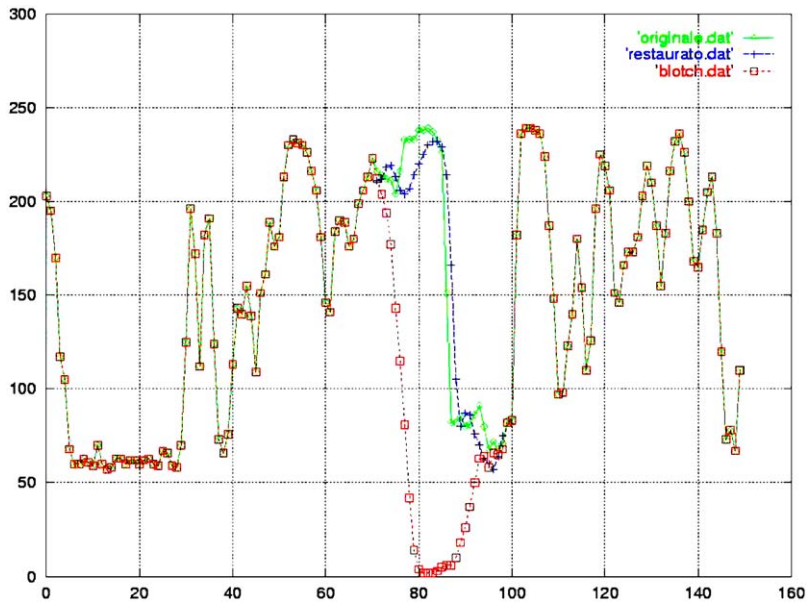

Fig. 8. Sequence *TreeD*—cross-section profile (neighbor of upper right blotch). The box line refers to the noisy data inside the blotch region, the star line refers to the restored brightness and the circle line indicates the original data.

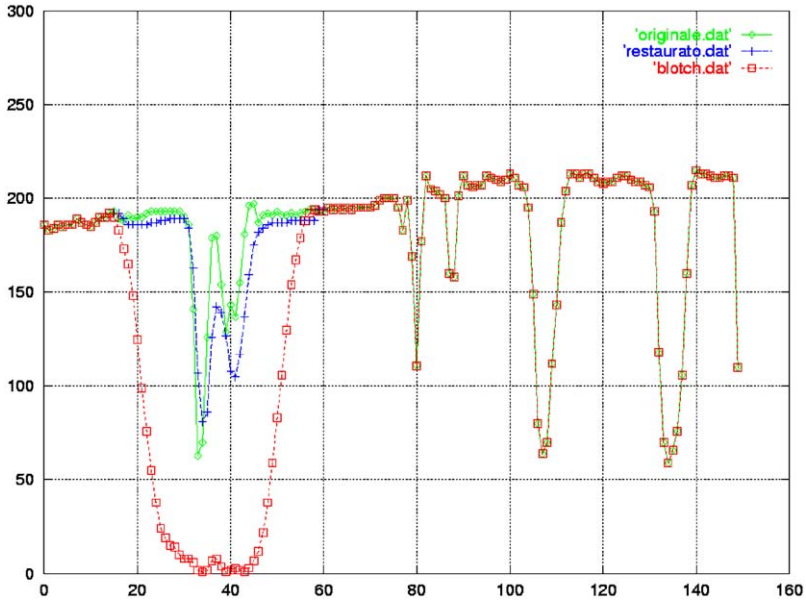

Fig. 9. Sequence *TreeD*—cross-section profile (neighbor of lower left blotch). The box line refers to the noisy data inside the blotch region, the star line refers to the restored brightness and the circle line indicates the original data.

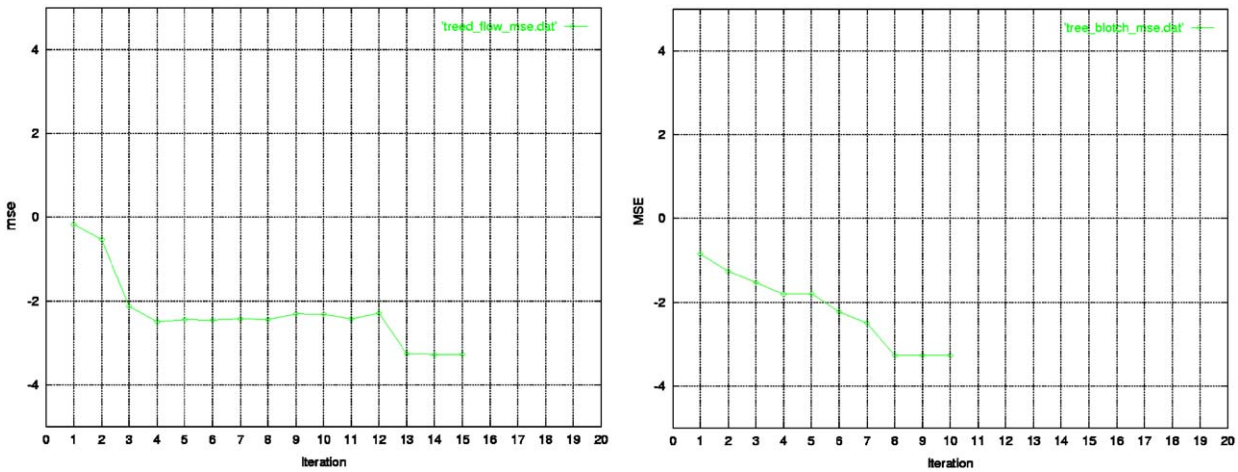

Fig. 10. Sequence *TreeD*—(sx) *optical flow*, (dx) *blotch removal*. MSE versus the iteration numbers.

Table 2
Angular error and standard deviation of the optical flow field—comparison with the most common regularization functionals as reported in [2]

Technique	Average error (in deg)	Std Dev (in deg)
Horn–Schunck	2.50	3.89
Lucas–Kanade	1.65	1.48
Uras	3.83	2.19
Nagel	3.21	3.43
Anandan	7.64	4.96
Singh	7.09	6.59
Feature matching	0.34	0.71
Our method	0.22	0.61

Fig. 11. Sequence *Naples*, size: 720 × 576 pixels.

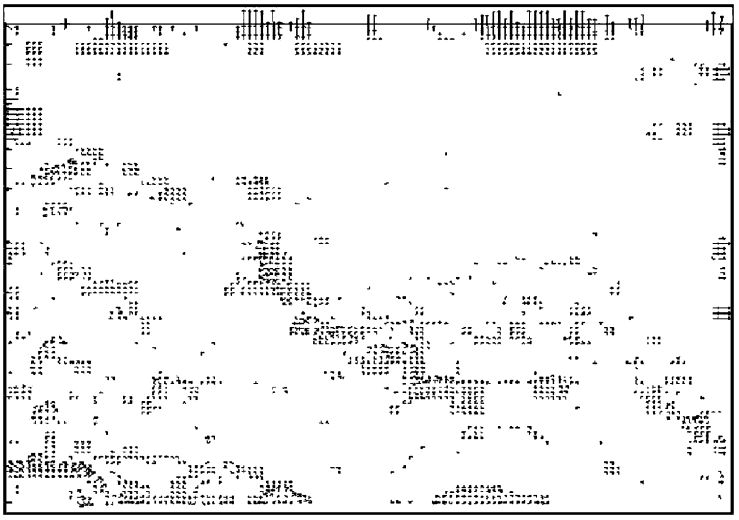

Fig. 12. Sequence *Naples*—optical flow. Cpu: 6.67 s. Options: level $= 0$, Sg $= 0.5$, St $= 0.5$.

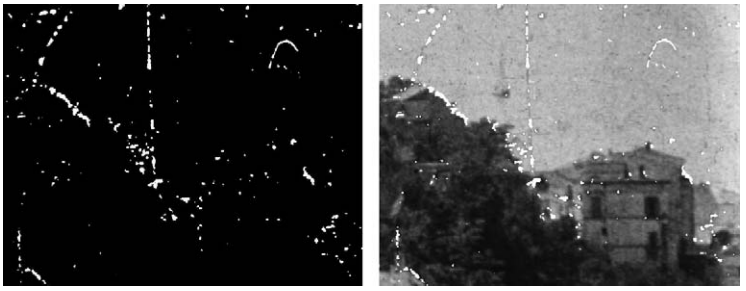

Fig. 13. Sequence *Naples*—on the left, the blotch mask. Options: $K = 60$. On the right the blotch mask overlapped to the corrupted frame.

The second test refers to the *Sequence Naples*; three frames of the *real movie* are shown in Fig. 11.

The sequence image is hardly noised, there are many small blotches as shown by the blotch mask in Fig. 14. The computed motion field, shown in Fig. 12, is nearly always translational. Restoration is shown in Figs. 13 and 14 (see also Fig. 15 for MSE). In this case we can just perform a qualitative evaluation of the restored image. However, the behavior of the numerical software on synthetic tests allows to extrapolate its performance in such cases giving a quality assurance also in real applications.

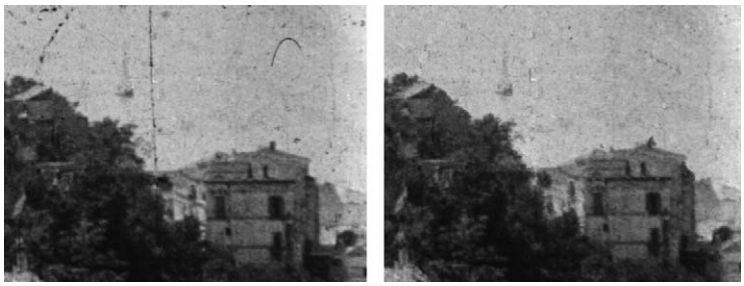

Fig. 14. Sequence *Naples*—(left) the original frame; (right) the restored frame. Cpu: 28.278 s.

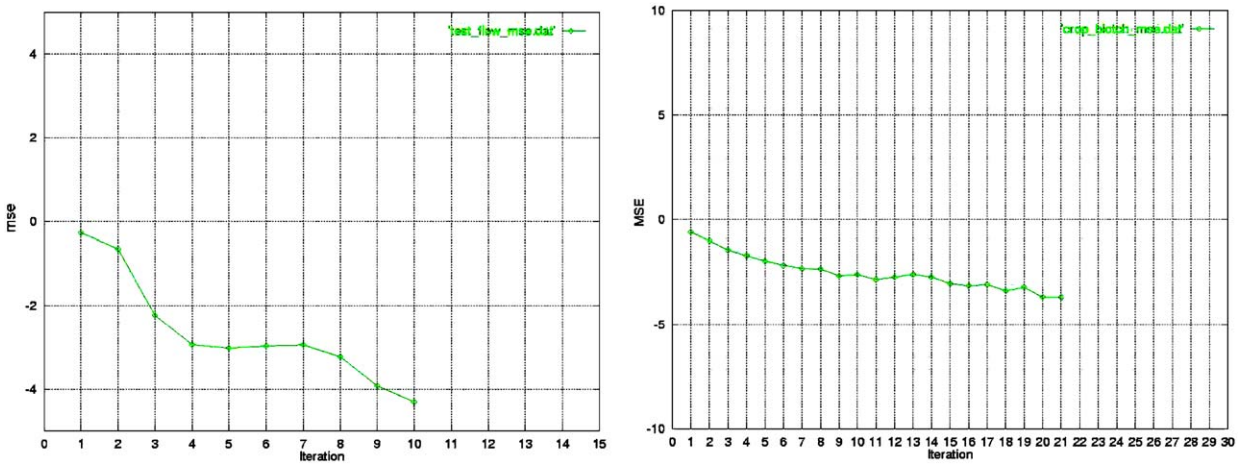

Fig. 15. Sequence *Naples*—(left) optical flow; (right) blotch removal. MSE versus iteration numbers.

7. Conclusion and future work

A key issue in developing application codes is to deal with the overall software complexity. This is even more true for imaging applications.

This work is the first step towards the development of a software tool to be actually used in digital film restoration. We describe computational efforts aimed at developing effective tools able to detect and remove a particular type of local random defect of old black-and-white films, commonly referred to as *blotch*.

To this aim we need to *detect the blotch domain* in an automatic way, to compute the *optical flow*, to *recover the brightness* inside the blotch domain. The numerical approach consists of an accurate motion estimation, based on a multigrid approach, and a two step algorithm: we first detect blotch locations by using motion-compensated and the temporal discontinuities that characterize the motion field in a corrupted sequence; then we remove the blotches, using information from the neighboring frames, and interpolating along motion trajectories.

Experimental results, equipped with several quality measures, illustrate the reliability and the robustness of the algorithm tested on both synthetic and real sequences.

References

[1] G. Aubert, P. Kornprobst, Mathematical Problems in Image Processing. Partial Differential Equation and the Calculus of Variations, Applied Mathematical Sciences, vol. 147, Springer, Berlin, 2002.

[2] J. Barron, D.J. Fleet, S.S. Bacuemin, Performance of optical flow techniques, Internat. J. Comput. Vision 12 (1994) 43–77.

[3] W.L. Briggs, A Multigrid Tutorial, SIAM, Philadelphia, PA, 1987.

[4] T.F. Chan, J.J. Shen, L. Vese, Variational PDE models in image processing, Notices of the AMS 50 (2002).
[5] P. Charbonnier, G. Aubert, L. Blanc-Fraud, M. Barlaud, Deterministic edge-preserving regularization in computed imaging, IEEE Trans. Image Process. 6 (1997) 298–310.
[6] J.P. Cocquerez, L. Chanas, J. Blanc-Talon, Simultaneous-inpainting and motion estimation of highly degraded video-sequences, SCIA, Lecture Notes in Computer Science, vol. 2749, Springer, Berlin, 2003, pp. 685–692.
[7] L. D'Amore, L. Marcellino, A. Murli, Un software numerico basato sull'approccio in multirisoluzione per la stima del flusso ottico, ICAR-TR-09-04, 2004.
[8] E. Decencière-Ferrandieère, Restauration Automatique de films anciens, Thèse de doctorat, Ecole Nationale Superieure des Mines de Paris, Dècembre 1997.
[9] B. Despas, F. Helt (Eds.), La restauration numerique des films cinmatographiques, Commission Suprieure et Technique de l'image et du Son, 1997.
[10] D. Geman, G. Reynolds, Constrained restoration and the recovery of discontinuities, IEEE Trans. Pattern Anal. Machine Intelligence 14 (1993) 367–383.
[11] B.K.P. Horn, B.G. Schunck, Determining Optical Flow, vol. 572, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, 1980.
[12] A.C. Kokaram, Motion picture restoration, Ph.D. Thesis, Department of Engineering, University of Cambridge, 1993.
[13] R.L. Lagendijk, P.M.B. Roosmalen, J. Biemond, Video Enhancement and Restoration, Faculty of Information Technology and Systems, Delft University of Technology, ND, 1999.
[14] F. Lauze, M. Nielsen, A variational algorithm for motion compensated inpainting, British Machine Vision Conference, 2004.
[15] L. Marcellino, Su alcuni metodi numerici per il restauro automatico digitale di immagini, Ph.D. Thesis, Universitá degli Studi di Napoli Federico II, 2004.
[16] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Trans. Pattern Anal. Machine Intelligence 12 (1990) 629–639.
[17] A. Rosenfeld, A.C. Kak, in: W. Rheinboldt (Ed.), Digital Picture Processing, vol. 1, Academic Press, New York, 1982.
[18] P. Scahllauer, A. Pinz, W. Haas, Automatic restoration algorithms for 35 mm film, Videre J. Comput. Vision Res. 1 (3) (1999).
[19] A.N. Tikhonov, V.Y. Arsenin, Solution of Ill-Posed Problems, Winston and Son, Washington, DC, 1997.
[20] J. Weickert, O. Scherzer, Relations between regularization and diffusion filtering, T.R. DIKU-98/23, Department of Computer Science, University of Copenhagen, Denmark, 1998.
[21] J. Weickert, C. Schnorr, et al., Variational optical flow computation in real time, Preprint no. 89, University of Saarland, Germany, 2003.