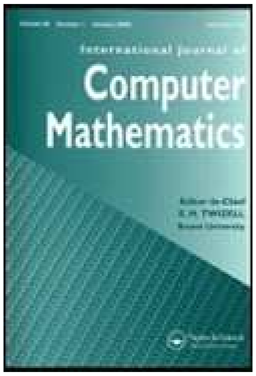
A numerical algorithm for image sequence inpainting that preserves fine-textures

Livia Marcellino, Luisa d'Amore, Ardelio Galletti, Daniela Casaburi



A numerical algorithm for image sequence inpainting that preserves fine-textures

Journal:	*International Journal of Computer Mathematics*
Manuscript ID:	GCOM-2010-0425-B.R2
Manuscript Type:	Original Article
Date Submitted by the Author:	26-Oct-2010
Complete List of Authors:	Marcellino, Livia; University of Naples Parthenope, Dept. of Applied Science d'amore, luisa; Department of Math. and Applications Galletti, Ardelio; University of Naples Parthenope, Dept. of Applied Science Casaburi, Daniela; SPACI
Keywords:	video inpainting, fine texture preservation, PDE-based models, GMRES iterative method, AMG preconditioner

International Journal of Computer Mathematics
Vol. 00, No. 00, Month 200x, 1–15

A numerical algorithm for image sequence inpainting that preserves fine-textures.

D. Casaburi[a], L. D'Amore[b], A. Galletti[c], L. Marcellino[c*]

[a]*SPACI - Naples, Italy*
[b]*University of Naples Federico II,*
Complesso Universitario di Monte S. Angelo, Via Cinthia - Naples, Italy
[c]*University of Naples Parthenope,*
Centro Direzionale, Isola c4 - Naples, Italy

(*Received 00 Month 200x; in final form 00 Month 200x*)

We describe a fast, reliable and automatic algorithm for image sequence inpainting that combines spatio-temporal interpolation with fine texture preservation inside missing areas. The algorithm provides an estimate of the inpainting error by using an automatic geometric recognition of missing regions. Computational kernels are sparse linear systems solved using GMRES iterative method equipped with AMG multigrid preconditioner. Experiments on synthetic and real data are discussed.

Keywords: video inpainting, fine texture preservation, PDE-based models, GMRES iterative method, AMG preconditioner.

AMS Subject Classification: 65M32, 68U10.

1. Introduction.

Image inpainting is an inverse and ill posed problem [2]. It concerns reconstruction of missing values of the image intensity function that appear as *local random* visual defects (gaps, scratches, holes,...). To handle the ill-posedness of such inverse problem, additional information is employed (regularization).
In recent years, many regularization methods have been proposed for image inpainting. Overall, two main approaches can be selected: those called *diffusion* models, relying on partial differential equations (PDE) and those called *texture synthesis* (TS) models, using statistical methods.
Diffusion models employ continuity of geometrical structures of images [2, 5]. These are mainly suitable for treating small-narrow gaps of piecewise smooth images, also called "cartoon" images, which have non-textured regions.
Texture synthesis techniques are introduced in order to recover textured natural images. These methods compute new instances of a texture from a smaller sample and are mainly used for images containing several textured areas [9, 12]. Texture synthesis technique has also been applied to inpaint features, as in [4], where the idea is to decompose first an image into a structured part and a textured part, then to apply different techniques separately to both parts. On the other side, in [11], the heuristic patches copy-paste technique presented in [12] has been formulated in a variational framework.
Other techniques such as morphological component analysis are also applied on simultaneous cartoon and texture image inpainting, as in [13].
Motivated by a large amount of work on image inpainting, some authors proposed to solve the inpainting problem in the wavelet domain. In [6], an efficient method to recover piecewise constant or smooth images by combining total variation regularization and wavelet representations is proposed. While, in [30], this idea is extended to nonlocal total variation regularization in order to recover textures and local geometry structures simultaneously.

A straightforward extension of these methods for *video inpainting* is to treat data as a set of distinct images and to restore them individually [3].
To take advantage of high temporal correlation of video sequence some of these algorithms use the property that characterizes local random defects, i. e. the spatial position of these varies significantly frame by frame. As a consequence, it is reasonable to assume that missing

[*] Corresponding author. Email: livia.marcellino@uniparthenope.it

ISSN: 0020-7160 print/ISSN 1029-0265 online

DOI: 10.1080/0020716YYxxxxxxxx
http://www.informaworld.com

areas can be modeled and localized as temporal discontinuities of intensity function [19]. Within diffusion methods based on PDE, that proposed in [23] uses local patches to fill the holes, by first separating foreground objects from background layer (using the motion field). Then, the holes in the moving foreground regions are inpainted by using a priority-based exemplar process. Damaged patches around the boundary of the hole are filled by selecting candidates from the foreground mosaic that minimizes a metric distance. This technique is not useful if a significant portion of the object is missing, because of the sharp smoothing, it does not reproduce the texture information and suffers from severe blurring artifacts.
A video inpaintig scheme based on motion compensation and TS-completion has been proposed in [29]. After removing a particular motion layer, motion compensation is used to complete moving objects and non-parametric texture synthesis is used to complete the static background regions. The inpainted layers are then warped into every video frame to complete the holes. While being effective for textured images these approaches are susceptible to growing incorrect patches due to spurious local variation.
Deviating from the patch methods discussed above, in [17] is introduced an object-based inpainting system which utilizes a user-assisted segmentation to inpaint holes in foreground regions that are characterized by a cyclic movement. To complete the missing foreground regions the periodicity of the moving foreground object is estimated and used to reconstruct them. Algorithms following this approach very often have higher performances because segmentation into different layers not only provides better matching results, but also significantly reduces the search space for finding appropriate matches used for inpainting.
In conclusion, even though some video inpainting algorithms have been proposed, many algorithmic topics still remain to investigate. Among them, there is the need to perform efficient computations, to provide reliable results and to reduce user's interactions.

Here, we propose a numerical algorithm aimed to reconstruct missing data in a video sequence by preserving fine structures. The goal is to suggest a convenient video inpainting model, based on motion estimate, and to combine efficient and reliable numerical approaches for texture-preserving image reconstruction.
Main contributions of this work are the employment of:

- a *reliable* motion estimate that assumes that spatial brightness gradient does not change over time. This model, proposed in [26], is suitable in presence of fine-textures.
- a *texture-preserving* discretization scheme, as suggested in [18]. In order to avoid or minimize the loss of important fine structures, *essentially non-dissipative* (ENoD) schemes are adopted.
- a *fast* solution of the underlying computational kernels. We use GMRES iterative method equipped with Algebraic Multigrid preconditioner to solve sparse linear systems.
- an *automatic* detection of corrupted regions and of their shapes to estimate the inpainting error. We implement the model proposed in [7].

The paper is organized as follows. In section 2, we introduce the mathematical models we are going to use and we show how to compute the error estimate. In section 3, we describe the numerical approach and main computational kernels. Finally, in section 4, numerical experiments on real and synthetic image sequence compared with graphs of cross-section are presented. Section 5 concludes the paper.

2. The motion-aided image sequence inpainting.

In this section we review preliminary definitions and introduce the mathematical models describing the image sequence inpainting problem.

[Def 1] Image sequence brightness function:
Let $J \subset \Re$ be a bounded interval. Given $t \in J$, let $P(t) \equiv (x(t), y(t)) \in \Omega$, where $\Omega = \Omega_x \times \Omega_y \subset \Re^2$ is the image plane [1]. The image sequence is defined as the piecewise differentiable function:

$$I : t \in J \mapsto P(t) \in \Omega \mapsto I(t) \equiv I(P(t), t) \in [0, 255]$$

[Def 2] Motion trajectory:
Given the image sequence $I(t)$, the motion trajectory is the line or arc of line L defined by successive positions of $P(t)$ as $t \in J$. The parametric equation for L is:

$$L : \begin{cases} \Delta x = x(t + \Delta t) - x(t) = \Delta t \;\; u(t) \\ \Delta y = y(t + \Delta t) - y(t) = \Delta t \;\; v(t) \end{cases}$$

where $(u(t), v(t))$ are the components of the motion vector, at each $t \in J$, and $\Delta x, \Delta y$ are spatial displacements.

[1] The image plane Ω should depend on the acquisition time t. In practice, it is the same at each t because it refers to the rectangular plane of the image acquisition. Then, for simplicity of notations, we obmit the dependence of Ω on t

[Def 3] The image sequence inpainting problem:
Let $I(t)$ be the noisy-free image sequence brightness function. Let b be the *characteristic function* which defines the corrupted area $B \subset \Omega$ and let $I_B(t)$ be the missing values of the intensity brightness function. Then, the function:

$$\widetilde{I}(P(t),t) = [1 - b(P(t),t)] \;\; I(P(t),t) + b(P(t),t) \;\; I_B(P(t),t)$$

denotes the image sequence brightness function to restore.
More precisely, given $\widetilde{I}(t)$, to restore the image sequence brightness function for each $t \in J$, we need to determine the *characteristic function* b and the *the missing data* $I_B(t)$ on $B \subset \Omega$. Moreover, to integrate temporal information, we compute the *motion field*.

[P1] Motion estimate:
Assuming that $\frac{d}{dt}\nabla I(t) = 0$, following [28], we compute the motion field, solving a couple of PDEs, with Dirichlet boundary conditions:

$$\begin{aligned} \tfrac{\partial}{\partial \tau} u(\tau,t) = \alpha_{OF} \cdot \nabla[\phi'(\nabla u \nabla u^T + \nabla v \nabla v^T)\nabla u] + \\ -2[I_{xx}u + I_{yx}v + I_{tx}] \cdot I_{xx} - 2[I_{xy}u + I_{yy}v + I_{ty}] \cdot I_{xy} \\ \tfrac{\partial}{\partial \tau} v(\tau,t) = \alpha_{OF} \cdot \nabla[\phi'(\nabla u \nabla u^T + \nabla v \nabla v^T)\nabla v] + \\ -2[I_{xx}u + I_{yx}v + I_{tx}] \cdot I_{yx} - 2[I_{xy}u + I_{yy}v + I_{ty}] \cdot I_{yy} \end{aligned} \qquad (1)$$

where τ is the scale-parameter, $(u_0(t), v_0(t)) = ((u(0,t), v(0,t)) = (0,0)$ are the initial conditions, when $\tau = 0$, α_{OF} is the regularization parameter and $\phi'(s^2) = \varepsilon + \frac{(1-\varepsilon)}{2\sqrt{1+s^2}}$ is the diffusivity function, with $\varepsilon = 10^{-3}$.

Using the motion field, we address the video inpainting problem using the following assumption: "*sequence brightness level is constant along the motion trajectory*", [16]. This means that directional derivatives of the brightness function $I(t)$, along the *positive* and *negative* directions of the motion trajectory are zero, i. e.:

$$\begin{aligned} \tfrac{\partial}{\partial L^+} I(t) = \lim_{\Delta t \to 0^+} \tfrac{I(x(t-\Delta t)+\Delta x^+, y(t-\Delta t)+\Delta y^+, t-\Delta t) - I(x(t),y(t),t)}{\Delta t} = 0 \\ \tfrac{\partial}{\partial L^-} I(t) = \lim_{\Delta t \to 0^-} \tfrac{I(x(t+\Delta t)-\Delta x^-, y(t+\Delta t)-\Delta y^-, t+\Delta t) - I(x(t),y(t),t)}{\Delta t} = 0 \end{aligned} \qquad (2)$$

However, at corrupted positions this property is not valid. Then, the inpainting regions and their shapes can be automatically localized checking temporal discontinuity of the intensity brightness functions.

[P2] Inpainting domain detection and shape recognition:
As in [10], we determine the inpainting domain $\mathring{B} \subset \Omega$, as follows:

$$\mathring{B} = \left\{ P(t) \in \Omega : \min \left\{ \left| \frac{\partial \widetilde{I}}{\partial L^+} \right|, \left| \frac{\partial \widetilde{I}}{\partial L^-} \right| \right\} \neq 0 \right\} \qquad (3)$$

Hence, for every $t \in J$:

$$b(P(t),t) = \begin{cases} 1 \;\; if P(t) \in \mathring{B} \\ 0 \;\; if P(t) \in \Omega \backslash \mathring{B} \end{cases} \qquad (4)$$

Using b we define the *inpainting mask* and its shape as $B = \mathring{B} \bigcup E$, where E is a fixed closed domain in $\Omega \backslash B$, defined in [5].

[P3] Missing data reconstruction:
Following [20], we compute I_B on the inpainting mask B, by solving the following PDE-model, with Neumman boundary conditions:

$$\frac{\partial}{\partial \tau} I_B(\tau,t) = |\nabla I_B| \cdot [I_C - I_B] + \alpha_I \nabla \cdot \left[\frac{\nabla I_B}{|\nabla I_B|} \right] |\nabla I_B| \qquad (5)$$

where τ is the scale-parameter, $|\nabla I_B| = \sqrt{(\partial I_B/\partial x)^2 + (\partial I_B/\partial y)^2}$, α_I is the regularization parameter and the function $I_C(t) = I_B(0,t)$ is the initial condition, when $\tau = 0$, defined as follows:

$$I_C(t) = \frac{|I_{t-\Delta t}(t) + I_{t+\Delta t}(t)|}{2}$$

where:

$$\begin{aligned} I_{t-\Delta t}(t) &= I(x(t-\Delta t)+\Delta x^{+}, y(t-\Delta t)+\Delta y^{+}, t-\Delta t) \\ I_{t+\Delta t}(t) &= I(x(t+\Delta t)-\Delta x^{-}, y(t+\Delta t)-\Delta y^{-}, t+\Delta t) \end{aligned}$$

Summarizing, we perform the image sequence inpainting using motion trajectory and its properties. The corrupted positions and their shapes can be localized as temporal discontinuities of the intensity brightness function. Then, restoration is performed using information from the previous and the next frame.
The overall algorithm consists of three successive steps:

- **P1**: optical flow computation.
- **P2**: inpainting mask detection and shape recognition.
- **P3**: missing data reconstruction.

Related dataflow is the following: $\forall t \in J$, $\widetilde{I}(t) \mapsto (u(t), v(t)) \mapsto b(t) \mapsto I_B(t)$.
It is worth to note that in [10], the authors followed the same three basic steps. However, main differences are in steps **P1** and **P2**. Here a different mathematical model for the motion estimate (P1) is adopted, moreover a different discretization scheme for data reconstruction (P2) is employed. All these changes deliver more accurate results than those reported in [10]. Comparisons will be shown in Section 4.

2.1 *Error estimate.*

Let I_B be the uncorrupted image brightness functions on B. If I_{B_R} denotes the solution of (5), at each point $P(t) = P(x(t), y(t)) \in B$, the *inpainting error* is defined as follows:

$$err\,[P(t), t] = \left| I_{B_R}(P(t), t) - I_B(P(t), t) \right|$$

Then, the error over B is bounded as follows:

$$err(B) = \int_B err\,[P(x(t), y(t)), t]\, dxdy \leq |B| \max_B \{err\,[P(x(t), y(t)), t]\}$$

where $|B|$ is the area of B.
Following [7], we have that:

$$err(B) \leq |B| \max_B |\nabla I_{B_R}(P(t), t)| \leq |B| M_t$$

where M_t is defined as:

$$M_t = \max_B \{|\Delta I_{B_R}(t)|\} = \max_B \left\{ \left| \frac{\partial^2}{\partial x^2} I_{B_R}(x(t), y(t), t) + \frac{\partial^2}{\partial y^2} I_{B_R}(x(t), y(t), t) \right| \right\}$$

We consider three kinds of shape of B:

1. **circle -** If the inpainting domain B can be covered by a circle C, with diameter d (see figure 1), we have $\forall (P(t), t) \in \Omega \times J$:

$$err\,[P(t), t] \leq \frac{M_t}{4} d^2, \forall t \in J \qquad (6)$$

2. **ellipse -** For a long and narrow region, the domain B can be covered by an ellipse E with β, α as the minor and the major diameter, respectively (see figure 1). In this case we have $\forall (P(t), t) \in \Omega \times J$:

$$err\,[P(t), t] \leq \frac{M_t}{8} \beta^2 \qquad (7)$$

3. **like-ellipse -** For any domain B, when the thickness is small but may have a complicated shape (see figure 2), the error bound given by (6) or (7) could be pessimistic. In this case, B is mapped to an ellipse-like domain $\widetilde{B_e}$, and we have $\forall (P(t), t) \in \Omega \times J$:

$$err\,[P(t), t] \leq \frac{\widetilde{M_t}}{8} \beta^2_{\widetilde{B_e}} \qquad (8)$$

where $\widetilde{M}_t$ is the bound computed on $\widetilde{B_e}$ and $\beta_{\widetilde{B_e}}$ is the minor diameter of ellipse $\widetilde{B_e}$.
Domain B can be mapped to a circle-like domain $\widetilde{B_c}$ too (see figure 2). However, in this case the error estimate could not reliable. Indeed, regions $\widetilde{B_e}$ and $\widetilde{B_c}$ have the same areas (in terms of number of pixels), but the diameter of the circle domain may be larger than the minor diameter of ellipse domain, i.e. $d_{\widetilde{B_c}} >> \beta_{\widetilde{B_e}}$ and:

$$\frac{\widetilde{M}_t}{8}\beta^2_{\widetilde{B_e}} << \frac{\widetilde{M}_t}{4}d^2_{\widetilde{B_c}}$$

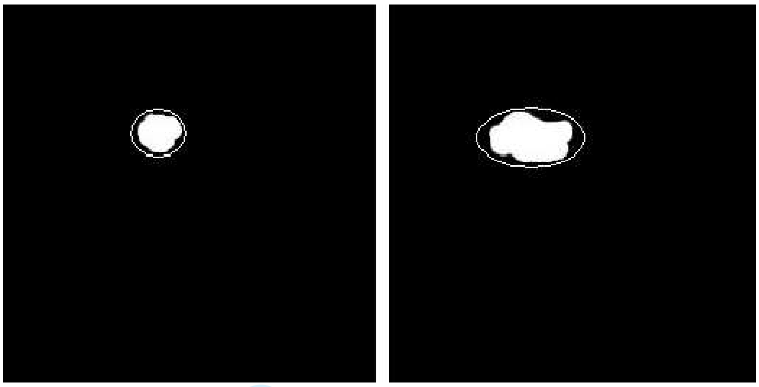


Figure 1. On the left the inpainting mask covered by a circle. On the right the inpainting mask covered by an ellipse

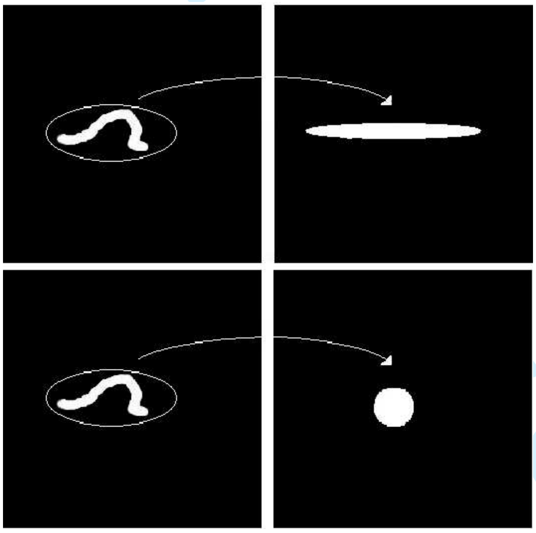


Figure 2. Up: on the left the inpainting mask small and with complicated shaped; on the right the ellipse-like transformed domain. Down: on the left the inpainting mask small and with complicated shaped;on the right the circle-like trasformed domain. The shapes are different, but they have the same pixel number.

3. Numerical Algorithm.

In this section we describe numerical approach and main computational kernels.
Let us give some notations:

- if $N \times N$ is the dimension of a frame and n the step number with respect to axes x and y (i.e. the number of pixels in both directions), then we denote by (x_l,y_m), the discrete space variables. Moreover, $x_{l+1} = x_l + h_x$ and $y_{m+1} = y_m + h_y$, where: $h_x = h_y = N/n$, $\forall l, m = 1, ..., n$.

– if nf is the number of frames, we denote by t_k the discrete time variables and by Δt the time step, so that $t_{k+1} = t_k + \Delta t, \forall k = 1, ..., nf$
– if $[0, T]$ is the scale interval and $nscales$ the number of scale steps, we denote by τ_i the ith scale-step $\forall i = 1, ..., nscale$, so that $\tau_{i+1} = \tau_i + \Delta\tau$, where $\Delta\tau = T/nscale$ is the step-size.

Therefore, we get the algorithm described in Table 1.

```
for $k = 1$ to $nf - 1$ do number of frames

    P1: optical flow computation from frames (k, k+1) and frames (k+1, k+2)
        - solution of PDEs, as defined in (1)

    P2: inpainting mask detection and shape recognition on frame k+1
        - discretization of directional derivatives as defined in (2)
        - computation of b as defined in (3) and (4)
        - shape recognition inside the inpainting mask

    P3: restoration of frame k+1
        - solution of PDE as defined in (5)
        - error estimate (6)-(8)

end for
```

Table 1. The overall automatic algorithm.

3.1 *Discretization.*

Concerning **P2**, directional derivatives, as defined in (2), are discretized using *finite difference* schemes. For every $k = 1, ..., nf$:

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

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

Then, as in [10], the algorithm computes the characteristic function $b, \forall k = 1, ..., nf$ as follows:

$$b(P(t_k), t_k) = \begin{cases} 1 \;\; if \min\left\{W_{t_k-\Delta t}(t_k), W_{t_k+\Delta t}(t_k)\right\} \geq K \\ \qquad\qquad 0 \;\; otherwise \end{cases}$$

where K is a given tolerance (we use $K \leq 50$).
Therefore, by using b the algorithm finds the inpainting mask and for each region of inpainting mask, detects both the maximum height and the maximum width and it compares them to determine if the region can be covered by a circle (6) or by an ellipse (7). If the height of the region is constant the bound defined in (8) is applied.
Finally, M_t is computed discretizing the laplacian operator $\Delta I_{B_R}(t)$ by using *central finite differences*.

Concerning discretization of **P1** and **P3**, it is desirable that the numerical scheme for scale-space discretization does not alter any image properties (it should be invariant with respect to many transformations: grey level shift, translation and rotation, etc...[1]).
We use the semi-implicit scheme because it meets these requirements as well as consistency, convergency and stability properties. Unlike the explicit scheme, this one is stable for all scale steps [27].
More precisely, we use the *forward finite difference* scheme for the scale derivative, so the non-linear diffusion PDE:

$$\frac{\partial}{\partial\tau} u(\tau) = |\nabla u|^2 \; \nabla \cdot \left[g(|\nabla u_\sigma|) \frac{\nabla u}{|\nabla u|^2} \right]$$

becomes:

$$\frac{u(\tau_{i+1})-u(\tau_i)}{\Delta\tau} = \nabla\cdot\left[g(|\nabla u_\sigma(\tau_i)|)\frac{\nabla u(\tau_{i+1})}{|\nabla u(\tau_i)|^2}\right]|\nabla u(\tau_i)|^2$$

where $\tau_{i+1} = \tau_i + \Delta\tau$.
For the space-time discretization of **P1**, we employ the *finite differences* scheme. Concerning **P3**, we use *finite differences* schemes for discretization of ∇I and *essentially non dissipative* scheme *of the second kind* (ENoD2) for discretization of $\nabla\cdot\left[\frac{\nabla I}{|\nabla I|}\right]$.
More precisely, to approximate the so-called edge-detector function, $|\nabla I(x,y)| = \sqrt{(\partial I(x,y)/\partial x)^2 + (\partial I(x,y)/\partial y)^2}$ we compute $|\nabla I(x_l, y_m)|$, following [18], as:

$$\min\{\left[(D^1 I(x_l,y_m))^2 + (D^2 I(x_l,y_m))^2\right]^{1/2}, \left[(D^3 I(x_l,y_m))^2 + (D^4 I(x_l,y_m))^2\right]^{1/2}\}$$

$\forall l, m = 1, ..., N$, where:

$$D^1 I(x_l,y_m) = \min\left[\frac{|I(x_l,y_m)-I(x_{l+1},y_m)|}{\sqrt{h_x^2,}}, \frac{|I(x_l,y_m)-I(x_{l-1},y_m)|}{\sqrt{h_x^2}}\right]$$

$$D^2 I(x_l,y_m) = \min\left[\frac{|I(x_l,y_m)-I(x_l,y_{m+1})|}{\sqrt{h_y^2,}}, \frac{|I(x_l,y_m)-I(x_l,y_{m-1})|}{\sqrt{h_y^2}}\right]$$

$$D^3 I(x_l,y_m) = \min\left[\frac{|I(x_l,y_m)-I(x_{l+1},y_{m+1})|}{\sqrt{h_x^2+h_y^2,}}, \frac{|I(x_l,y_m)-I(x_{l-1},y_{m-1})|}{\sqrt{h_x^2+h_y^2}}\right]$$

$$D^4 I(x_l,y_m) = \min\left[\frac{|I(x_l,y_m)-I(x_{l-1},y_{m+1})|}{\sqrt{h_x^2+h_y^2,}}, \frac{|I(x_l,y_m)-I(x_{l+1},y_{m-1})|}{\sqrt{h_x^2+h_y^2}}\right]$$

with: $x+1 = x+h_x, x-1 = x-h_x, y+1 = y+h_y, y-1 = y-h_y$, and h_x, h_y the step size along x and y, respectively.

ENoD2 discretization scheme takes into account of local structure of image discontinuities better than finite difference schemes. Indeed, differences of the intensity function I are computed along four directions (see figure 3) (vertical, horizontal and 45^o line segments). This approach allows to more accurately detect fine details such textures, avoiding Gibbs-like phenomenon obtained using centered schemes based on Taylor series in approximating functions with jump discontinuities. This is the original idea behind the ENO (essentially non oscillatory) interpolation schemes, initially introduced in [14], as well as all the upwinding techniques commonly used to discretize the level set equations . Comparison with results obtained in [10] using finite difference schemes are shown in section 4.

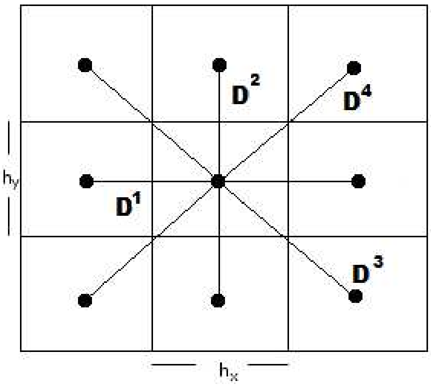


Figure 3. ENoD stencil.

3.2 *Computational kernels.*

Regarding **P1**, if the acquisition consists of nf frames and each image size is $N \times N$, we have to solve two linear systems, at each scale step $\forall i = 1, ..., nscale; \forall k = 1, ..., nf$:

$$\begin{array}{l} A(u(\tau_i, x_l^k, y_m^k), v(\tau_i, x_l^k, y_m^k)) \;\; u(\tau_{i+1}, x_l^k, y_m^k) = b(u(\tau_i, x_l^k, y_m^k), v(\tau_i, x_l^k, y_m^k)) \\ A(u(\tau_i, x_l^k, y_m^k), v(\tau_i, x_l^k, y_m^k)) \;\; v(\tau_{i+1}, x_l^k, y_m^k) = b(u(\tau_i, x_l^k, y_m^k), v(\tau_i, x_l^k, y_m^k)) \end{array}$$

where: $x_l^k = x_l(t_k), y_m^k = y_m(t_k)$ and with the same matrix $A \in \Re^{N^2 \times N^2}$, which is a block pentadiagonal matrix, with tridiagonal blocks along the main diagonal and diagonal blocks along the upper and lower diagonals.
Similary, for **P3**, at each scale step τ_i, the algorithm solves the linear system:

$$H(I(\tau_i, x_l^k, y_m^k)) \;\; I(\tau_{i+1}, x_l^k, y_m^k) = b(I(\tau_i, x_l^k, y_m^k))$$

with a sparse matrix $H \in \Re^{M \times M}$, where $M << N^2$, is the size of inpainting mask.
H is block pentadiagonal, with tridiagonal blocks along the main diagonal and diagonal blocks along the upper and lower diagonals.
We solve these linear systems employing the Generalized Minimum RESidual (GMRES) iterative method. GMRES is a Krylov subspace method designed to solve nonsymmetric linear systems [25]. In order to accelerate the convergence rate, we use the Algebraic Multigrid (AMG) as preconditioner equipped with the FALGOUT -CLJP coarse grid selection. This means that at each step of the GMRES method we need to apply the AMG multilevel cycle to update the residual. The central idea of multigrid method is to remove smooth components of the error, that cannot be removed by relaxation on a fine grid, by coarse-grid corrections. This is performed by using as relaxation method a standard iterative method. While geometric multigrid approach operates on predefined grid hierarchies depending on the domain problem, for the algebraic multigrid the necessary components for the hierarchical algorithm such as the coarse system matrices and the transfer operators are artificially created only from information contained in the algebraic equations. Definition of these components in AMG is done in a separate preprocessing step known as setup phase . Regarding the coarsening selection, which is the main ingredient of AMG, we adopt the FALGOUT-CLJP scheme because this is an hybrid approach combining the standard Ruge-Stuben (RS) method in the interior of the domain while using the Cleary-Luby-Jones-Plassman (CLJP) graph-partitioning coarsening on the boundaries. FALGOUT-CLJP exhibits the highest performance both in terms of convergence factor and of operator complexity (detailed explanations are in [15, 24]).

```
if $k = M$, set $x^k = (A^M)^{-1} b^M$
otherwise
    relax $A^k x^k = b^k$
    perform coarse grid correction:
        set $x^{k+1} = 0$, $b^{k+1} = I_k^{k+1}(b^k - A^k x^k)$
        solve on level $k + 1$
        correct the solution $x^k = x^k + I_{k+1}^k x^{k+1}$
    relax $A^k x^k = b^k$
```

Table 2. algorithm at level k.

3.3 *Computational cost.*

At each scale step and for each frame, the computational cost for solving **P1** is:

$$T_{OF}(N^2) = O(2k_{OF}5N^2) \;\; \forall \tau_i, t_k$$

For **P3**, we have:

$$T_I(M) = O(5k_I M) \;\; \forall \tau_i, t_k$$

k_{OF} and k_I are the iterations numbers of the preconditioned GMRES for solving **P1** and **P3**, respectively.

To state the performance of AMG preconditioner, we compare its execution time with that obtained using the classical Block Jacobi (BJ) preconditioner. BJ is defined by considering the block diagonals of the system matrix. We appreciate the rapid convergence of GMRES when equipped with AMG preconditioner instead of BJ. For instance, regarding P1, it reaches the residual accuracy of about $O(10^{-6})$ in 3 iterations and after 72 secs when using AMG preconditioner. While, using BJ, residual stagnates. See Table 3 and figure 4. Computing platform is a dual core Intel Processor i3-330UM at 1.2 GHz.

	GMRES+AMG	GMRES+BJ
P1	Total time: 71.21 secs. Nr of steps: 3 [1] 19.04 secs. [2] 15.56 secs. [3] 14.34 secs.	Total time: 148.61 secs. Nr of steps: 22 [1] 11.3 secs. ... [22] 5.32 secs.
P3	Total time: 9.83 secs. Nr of steps: 3 [1] 3.87 secs. [2] 1.86 secs. [3] 0.88 secs.	Total time: 16.34 secs. Nr of steps: 5 [1] 2.796 secs. [2] 2.456 secs. [3] 1.98 secs. [4] 1.39 secs. [5] 1.22 secs.

Table 3. Running time of GMRES+AMG including the setup time, and of GMRES+BJ for solving P1 and P3, respectively. At each row, the number inside parenthesis refers to each iteration number. Image size: 480 × 480.

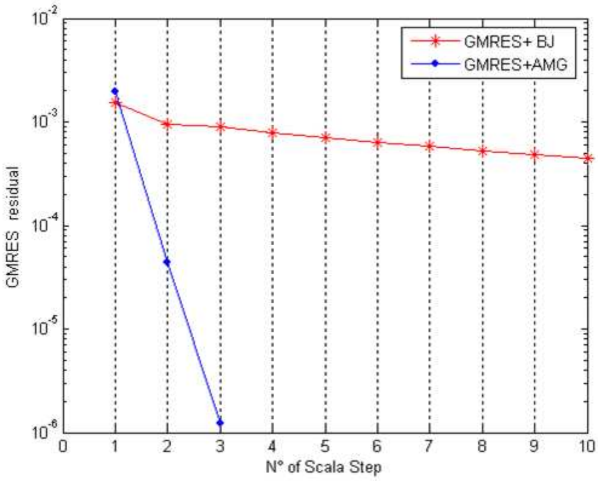


Figure 4. Numerical solution of P1. GMRES+AMG convergence behavior versus iteration number and first 10 iterations of GMRES+BJ.

4. Experimental results.

We carried out experiments aimed to verify accuracy and efficiency of the proposed approach. To this aim we consider real and synthetic sequences.
Here we show two tests: the first one concerns a simulated corrupted image sequence that we denote by *Barbara*, where missing data and motion (a zoom of the camera) are artificially generated. The original image *Barbara* is a real image. It contains both a texture and a cartoon part and is widely used in the image processing literature, as in [2, 5, 6, 9].
The second one refers to a real sequence of an old movie about *Naples*.

Below are shown the error estimate and the average error defined as:

$$A_{err} = \frac{1}{M}\sum_{B} |I_B - I_{B_R}|$$

where M is the number of pixels of the inpainted mask and I_B, I_{B_R} are the original uncorrupted intensity brightness function and the restored one, respectively. Results will be compared with those obtained in [10].

- **TEST 1:** *Barbara*
  Three frames of the sequence *Barbara* are shown in figure 5.

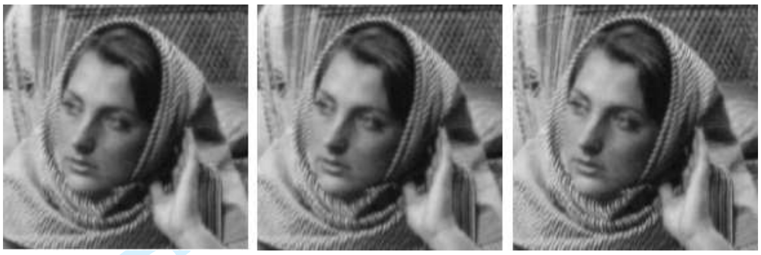


Figure 5. *Barbara sequence*: the motion, artificially generated, is a zoom of camera Number of frames $n_F = 3$, image size $N \times N = 190 \times 190$.

We reconstruct the middle image, corrupted by three *blotches* (black spots) artificially created on the face (where there is not a strong texture) and on the scarf (with texture). The corrupted image and the computed mask are shown in figure 6, while figure 7 shows results. Error estimate and cross section profile are shown in figures 8, 9, and 10.

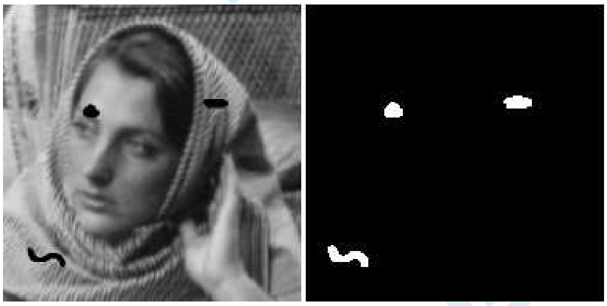


Figure 6. *Barbara sequence*: on the left the corrupted image (the blotches are made artificially); on the right the computed inpainting mask. Size of computed inpainting mask $M = 383$ pixels.

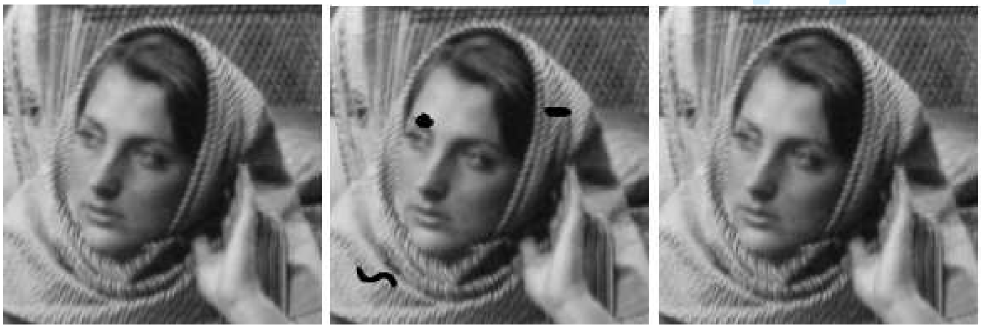


Figure 7. *Barbara sequence*: on the left the original uncorrupted image, in the middle the corrupted image, on the right the restored image. Optical flow computation: step-size of scale parameter $\Delta\tau_{OF} = 10^3$, regularization parameter $\alpha_{OF} = 10^{-4}$. Inpainting: step-size of scale parameter $\Delta\tau_I = 10$, regularization parameter $\alpha_I = 10^2$.

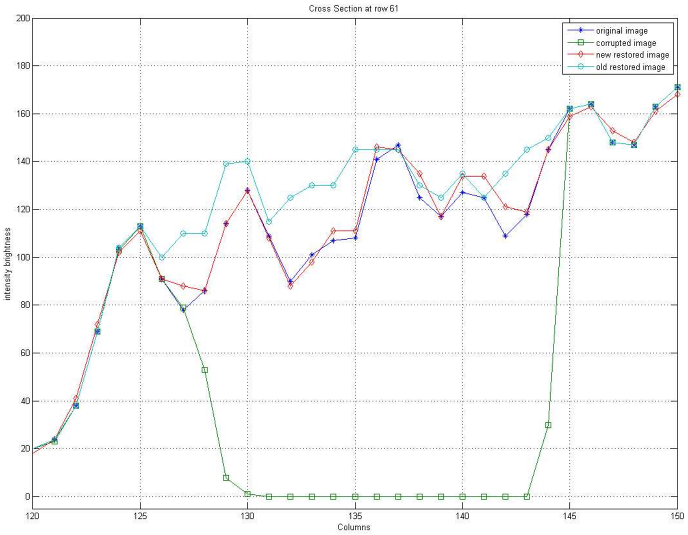


Figure 8. *Barbara sequence*: cross-section profile at row 61 i. e. at ellipse hole on the top right, with major diameter $\beta = 8$ pixels. $M_t = 0.21 \times 10^2$. Error bound estimate: $err = 0.17 \times 10^{-1}$. Average error $A_{err} = 0.39 \times 10^{-2}$. Previous average error $= 0.92 \times 10^{-1}$.

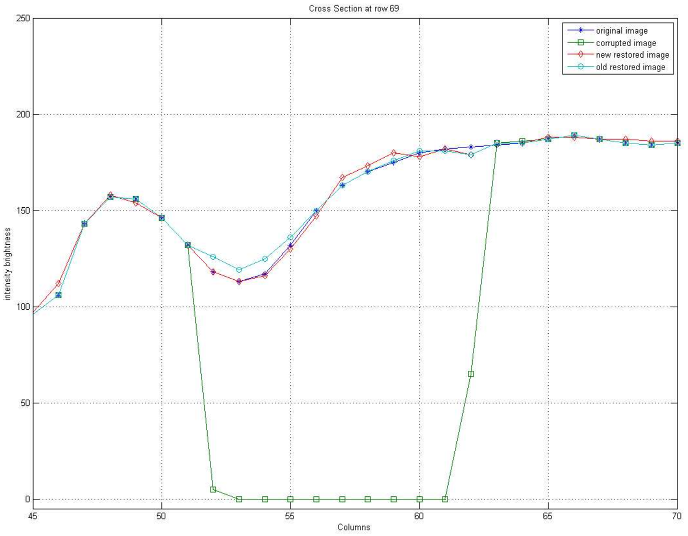


Figure 9. *Barbara sequence*: cross-section profile at row 69. i. e. at the circle hole in the middle, with diameter $d = 10$ pixels. $M_t = 0.13 \times 10^2$. Error bound estimate: $err = 0.34 \times 10^{-1}$. Average error $A_{err} = 0.41 \times 10^{-3}$. Previous average error $= 0.63 \times 10^{-3}$.

- **TEST 2:** *Naples*
  Three frame of the real movie before and after a denoising preprocessing are shown in fig. 11. The motion is nearly translational and often not uniform, therefore it is very hard to calculate. There are many small and large blotches as shown by the inpainting mask in figure 12. In this case we can just perform qualitative evaluation of the restored image (see figure 12). Error estimate and the cross section profile are shown in figures 13 and 14.

Regarding the *Barbara sequence*, cross-section profiles show that the lines corresponding to restored intensity brightness, using the new approach, always overlap that of the original intensity. Moreover, the average error always is smaller than that obtained without employing a texture preservation in [10]. The difference is more evident at the holes where the texture dominates, such as on the top right hole of *Barbara sequence* reconstructed in figure 8. In case of the real movie, *Naples*, reconstruction appears to be clearer than that

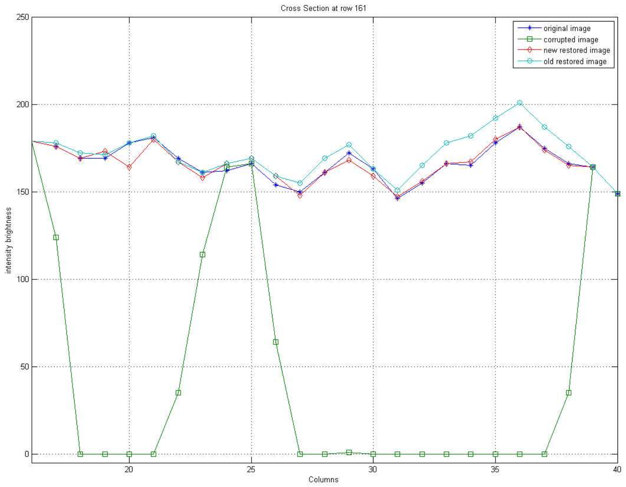


Figure 10. *Barbara sequence*: cross-section profile at row 161,i.e. at the bottom left like-ellipse, with major diameter $\widetilde{\beta} = 4$ pixels. $\widetilde{M}_t = 0.23 \times 10^{-2}$. Error bound estimate: $err = 0.1 \times 10^{-2}$. Average error $A_{err} = 0.58 \times 10^{-3}$. Previous average error $= 0.97 \times 10^{-3}$.

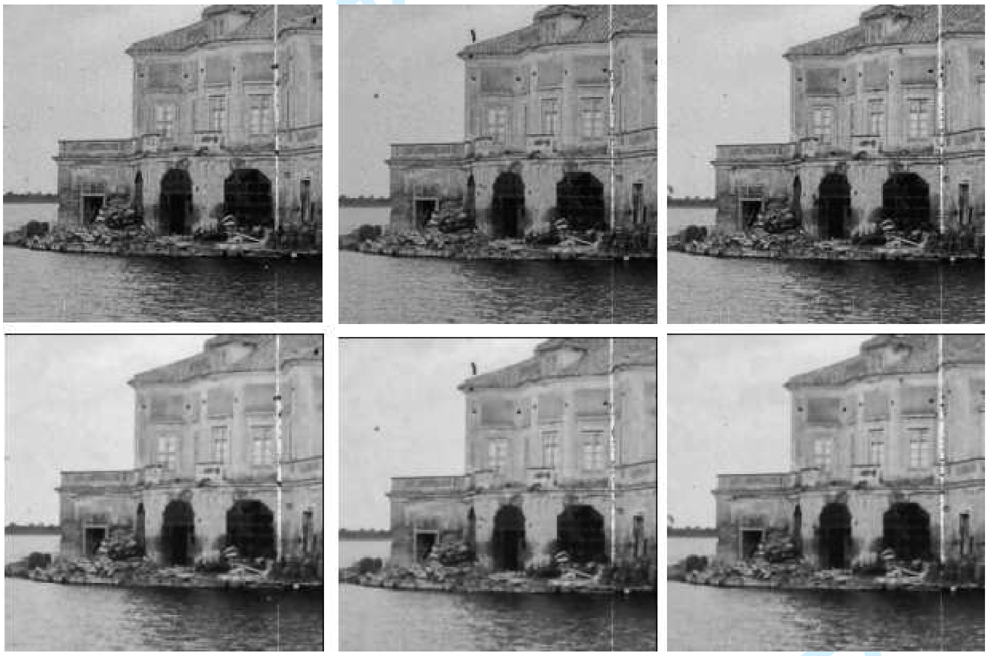


Figure 11. Three frames of the real sequence *Naples*. Up the original hardly noised sequence. Bottom the sequence after the denoising process. Number of frames $n_F = 30$, image size $N \times N = 480 \times 480$.

obtained in [10]. Indeed, looking at the cross-section profiles, we note that intensity values of the horizontal lines corresponding to the restored image, are greater than that obtained in [10]. This confirms that ENoD discretization takes into account the intensity of the neighborhood pixel (in this case, the foggy sky) instead of the straightforward interpolation performed by using the scheme in [10].
To state the efficiency of this algorithm, in tables 4 and 5, we report the running time of the whole three-step reconstruction, compared with those obtained in [10].

As expected, the computing time slightly increases with respect the previous runs because of the higher computational complexity of P1 and P3. However, the amount is not so much (just few seconds) thanks to the efficiency of AMG preconditioner.

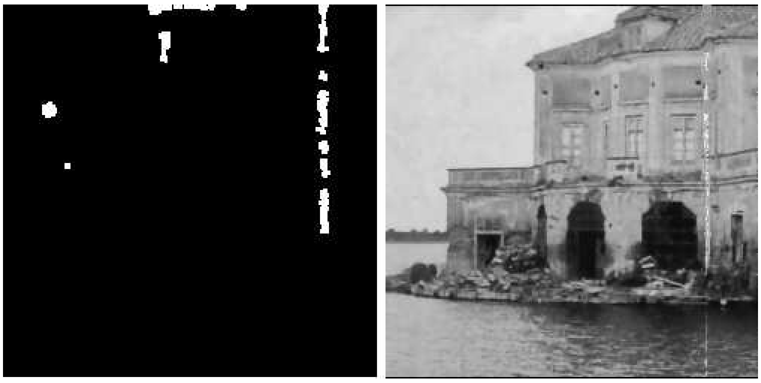


Figure 12. Real sequence *Naples*. On the left the computed inpainting mask, size $M = 2198$ pixels. On the right the restored frame. Optical flow computation: step-size of scale parameter $\Delta\tau_{OF} = 10^3$, regularization parameter $\alpha_{OF} = 10^{-6}$. Inpainting: step-size of scale parameter $\Delta\tau_I = 10$, regularization parameter $\alpha_I = 10^2$.

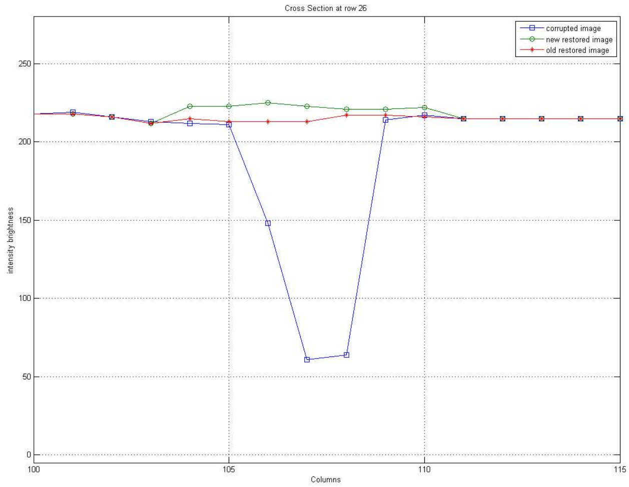


Figure 13. Real sequence *Naples*: cross-section profile at row 26. i. e. at the ellipse in the middle, with major diameter $\beta = 6$ pixels. $M_t = 0.12 \times 10^{-3}$. Error bound estimate: $err = 0.54 \times 10^{-3}$.

	video inpainting with textures	video inpainting image without textures
P1	12.31 secs. - [AMG setup: 0.81 secs]	10.702 secs.
P2	0.45 secs.	0.62 secs.
P3	4.62 secs. - [AMG setup: 0.93 secs]	2.64 secs.

Table 4. **Barbara sequence:** 190×190

5. Conclusion.

We describe a numerical algorithm for a fast, reliable and automatic video inpainting. It can effectively handle large regions of occlusion or missing data, combining a spatio-temporal interpolation with a fine texture preservation and provides an estimate of the inpainting error by using an automatic geometric recognition of missing regions.
The reconstruction relies on the motion trajectory and on its properties: the corrupted positions and their shapes can be localized as temporal discontinuities of the intensity brightness function while restoration is performed using information from the previous and the next frame.
The overall algorithm consists of optical flow computation, assuming that spatial brightness gradient does not change over time, inpainting mask detection and shape recognition,

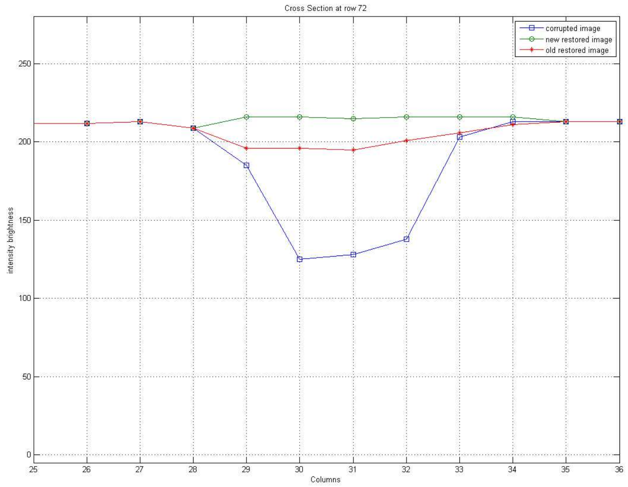


Figure 14. Real sequence *Naples*: cross-section profile at row 72. i. e. at the up left circle, with diameter $d = 10$ pixels. $M_t = 0.12 \times 10^{-3}$. Error bound estimate: $err = 0.3 \times 10^{-2}$.

	video inpainting with textures	video inpainting image without textures
P1	71.21 secs. - [AMG setup: 11.62 secs]	58.33 secs.
P2	1.78 secs.	1.81 secs.
P3	9.83 secs. - [AMG setup: 0.51 secs]	6.35 secs.

Table 5. **Naples sequence:** 480×480

showing that the error depends essentially on the geometry and on the shape of the domain, instead of the size of total area, and finally, the missing data reconstruction, where we use a texture-preserving discretization scheme - the essentially non-dissipative (ENoD) scheme. Solution of the underlying computational kernels is made by using GMRES iterative method equipped with Algebraic Multigrid preconditioner. Experimental results, equipped with several quality measures, illustrate the reliability and the robustness of the algorithm tested on synthetic and real sequences.
Comparisons with previous results demonstrate the better reliability of the reconstruction at the expense of a slight increase of the overall execution time.

References

[1] L. Alvarez, P. L.Lions, J. M. Morel - *Image selective smoothing and edge detection by nonlinear diffusion*. SIAM J. Num. Analysis. Vol 29. pp: 845-851, 1992.

[2] M. Bertalmio, G. Sapiro, V. Caselles, C. Balleste - *Image Inpainting*. Proceeding of SIGGRAPH 2000, New Orleans, 2000.

[3] M. Bertalmio, A. Bertozzi, G. Sapiro - *Navier-Stokes, fluid dynamics, and image and video inpainting.* Proceedings of International Conference on Computer Vision and Pattern Reecognition (CVPR). Vol. 1, Hawaii, pp. 355-362, 2001.

[4] M. Bertalmio, L. Vese, G. Sapiro and S. Osher - *Simultaneous structure and texture image inpainting*, IEEE Trans. Image Process, 12, 882?889, 2003.

[5] T. F. Chan, L. Verse, J. Shen - *Mathematical model for local deterministic inpaintings.* SIAM Journal of Applied Math. Vol 62(3), pp. 1019-1043, 2001.

[6] T. F. Chan, J. Shen and H-M Zhou - *Total variation wavelet inpainting.* J. Math. Imaging Vision, 25, pp. 107-125, 2006.

[7] T. F. Chan, S. H. Kang, - *Error Analysis for Image Inpainting*, Journal of Mathematical Imaging and Vision, 2006.

[8] A. Clery, R. falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. Mccornick, G. N. Miranda, J. W. Ruge *Robustness and scalability of algebraic multigrid.* SIAM J. Sci.Comp., Vol. 21 (5), pp. 1886-1908, 1999.

[9] A. Crimisi, P. Perez, K. Toyama - *Recognition filling and object removal by exemplar-based inpainting.* IEEE Transaction on Image Processing. Vol 13 (9) pp. 1200-1212, 2004.

[10] L. D'Amore, L. Marcellino, A. Murli - *Image sequence inpainting: towards a numerical software for*

detection and removal of local missing data via motion estimation. Journal of Computational and Applied Mathematics, 198, pp. 386-413, 2007.

[11] L. Demanet, B. Song and T. Chan - *Image inpainting by correspondence maps: A deterministic approach*, in Proc. VLSM conf., Nice, 2003.

[12] A. Efros, T. K. Leung - *Texture synthesis by non-parametric sampling.* IEEE International conference on Computer Vision (ICCV), Greece, pp. 1033-1038, 1999.

[13] M. Elad, J. Starck, P. Querre and D. Donoho - *Simultaneous cartoon and texture image inpainting using morphological component analysis* (MCA), Appl. Comput. Harmon. Anal., 19, pp. 340?358, 2005.

[14] A. Harten, B. Enquist, S. Osher and S. Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes III*, J. Comp. Phys. 71, pp. 231-303, 1987.

[15] Van E. Henson, U.M. Yang - *BoomerAMG: a Parallel Algebraic Multigrid Solver and Preconditioner.* Applied Numerical Mathematics, 41, pp. 155-177, 2002.

[16] B. K. P. Horn, B. G. Schunck - *Determining Optical Flow.* Artificial Intelligence Laboratory, Massachusetts Institute of Technology. Vol 572, 1980..

[17] J. Jia, Y.-W. Tai, T.-P. Wu, Tang - *Video repairing under variable illumination using cyclic motions.* IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI). Vol. 28, pp. 832-839, 2006.

[18] S. Kim - *Numerical modeling for the Recovery of fine structure in PDE-based image denoising*, Image and Vision Computing, 2008

[19] A. C. Kokaram - *Motion picture restoration.* Ph.D. Thesis, Department of Engineering, University of Cambrige, 1993.

[20] A. Marquina, S. Osher - *Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal*, SIAM J. Sci. Comput., 22 (2000), pp. 387-405.

[21] K. Mikula, A.Sarti, F.Sgallari - *Co-volume level set method in subjective surface based medical image segmentation*, in: Handbook of Medical Image Analysis: Segmentation and Registration Models (J.Suri et al., Eds.), Springer, New York, 2005, pp. 583-626l.

[22] S. Osher and R.P. Fedkiw - *Level set methods: an overview and some recent results*, J. Comput. Phys., 169(2). pp:463-502, 2001.

[23] K. A. Patwardhan, G. Sapiro, M. Bertalmio - *Video inpainting under constrained camera motion.* IEEE Transactions On Image Processing. Vol 16(2), pp. 545-553, 2007.

[24] J. W. Ruge, K. Stuben, *Algebraic multigrid (AMG)* - Frontiers in Applied Mathematics (SIAM, Philadelphia, 1987), vol. 3, pp. 73-130.

[25] Y. Saad, M. Schultz - *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856-869.

[26] S. Uras, F. Girosi, A. Verri, V. Torre, *A computational approach to motion perception* - Biological Cybernetics, Vol 60. pp. 79-87, 1988.

[27] J. Weickert *Recursive Separable Schemes for Nonlinear Diffusion Filters*, Scale Scace Theory in Computer Vision, Ed. by Romey, L. Florack, J. Koendrick, M. Viergever, LNCS 1252, Springer, pp. 260-271, 1997.

[28] J. Weickert, *On Discontinuity-Preserving Optic Flow*, Proc. Computer Vision and Mobile Robotics Workshop, pp.115-122, 1998.

[29] Y. Zhang, J. Xiao, M. Shah - *Motion layers based object removal in videos.* Proceedings of the Seventh IEEE Whorkshops on Application of Computer Vision. Vol 1, pp. 516-521, 2005.

[30] X. Zhang, T. F. Chan - *Wavelet Inpainting by nonlocal Total Variation.* Inverse Problem and Imaging. Vol. 4(1), pp. 191-210, 2010.