PAPER

Grey-level intensity measurements processing by means of Volterra equations and Least Squares Method for Video restoration

To cite this article: E Cuesta *et al* 2023 *Phys. Scr.* **98** 054002

Physica Scripta

PAPER

Grey-level intensity measurements processing by means of Volterra equations and Least Squares Method for Video restoration

E Cuesta[1,*], J Finat[2] and J Sánchez[3,4]

[1] University of Valladolid, E.T.S.I. of Telecomunication, Campus Miguel Delibes, Department of Applied Mathematics, Valladolid Spain
[2] University of Valladolid, Faculty of Sciences, Campus Miguel Delibes, Department of Algebra, Mathematical Analysis, Geometry and Topology, Valladolid Spain
[3] University Complutense of Madrid, Faculty of Sciences, Madrid, Spain
[4] Student
[*] Author to whom any correspondence should be addressed.

E-mail: eduardo.cuesta@uva.es, franciscojavier.finat@uva.es and jsanchezciria@gmail.com

Keywords: video restoration, Volterra equations, Mean Curvature Flow, Least Square Method

RECEIVED
16 November 2022

REVISED
16 March 2023

ACCEPTED FOR PUBLICATION
4 April 2023

PUBLISHED
18 April 2023

Abstract

In this paper a video sequence is considered as a two dimensional time evolving process. Under this assumption two Volterra equations based mathematical models are introduced for video restoration purposes. The first one is based on geometric features related to the spatial–time structure of the video sequence and gives rise to a nonlinear Volterra equation. This model arises from the Mean Curvature Flow linked to evolving surfaces. The second one is based on analytic features and leads to the formulation of a linear Volterra model. The second procedure relies on the assumption of local time coherence in the sequence of frames, at least short back in time. In both cases Volterra equations based approach introduces a memory effect in the process which in the present terminology means that several frames back may be taken into account for the better reconstruction of the current frame throughout a convenient choice of the convolution kernel. On the other hand the role played by Least Squares Method focuses on the practical computation of that convolution kernel just at discrete level. The performance of both approaches is shown through a list of suitable experiments, and the better performance of the second approach is illustrated with remarkable improvements in critical cases.

1. Introduction

Video processing is nowadays one of the most active fields of scientific research in Computer Vision, involving areas like superesolution [1], compression [2], or restoration [3] among others. The present paper focuses on the last point, and it is mainly focused towards the reconstruction of old movies which have been degraded, along the storage process or their conversion to digital format, by artifacts of different type. The most common artifacts are probably those related to the noise generation, the formation of blotches and scratches, the so–called Vinegard Syndrome on several frames, or Moiré and intensity Flicker effects (see [2, 4] and references therein). In [3] readers can find a detailed state–of–the–art about these topics at the early years of this century.

To fix ideas, we restrict ourselves to gray–level images since our proposals can be extended in a very simple way to the corresponding color representations. Every digital gray–level image can be understood as a discrete surface arising from a map $z = u(x, y)$, where u stands for the gray–level intensity. Hence, a digital gray–level video mini–sequence is understood as an one–parameter family of discrete surfaces arising from a family of maps $\{z = u(x, y, t)\}_{t\geqslant 0}$. Thus, its time evolution is understood as a time discrete evolutionary process. Notice that if in a set of pixels of a single frame of the sequence a large gray intensity change comes suddenly to appear (beyond some threshold), then these pixels are candidate to be damaged pixels.

In restoration techniques the pixels to be restored are typically are affected by irregularly distributed noise or by artificial blotches (i.e. unexpected dark or bright spots). There are very different denoising techniques for noisy images. In fact, evolving surfaces $z = u(x, y, t)$ representing the frames may show high gray–level variability

and different kinds of noise. Noise removal is actually out of the scope of this paper. Instead in this work we are certainly concerned with restoration of regions which are not well defined, where the information is lost, or, for instance, where higher resolution is required. Therefore advanced techniques of video enhancement and restoration are required.

The *main goal of video restoration* is to recover the original spatial and temporal structure by taking into account local information in the current, precedent, and even subsequent frames (if available). The recovery can be performed in statistical terms (by using correlations, e.g.), analytical terms (weighted averages based on convolution operators, e.g.) or geometric terms (by using e.g. conformal flows or, alternately, curvature flows for the underlying surfaces among others). The choice of the most appropriate approach depends on evolving geometric versus radiometric characteristics of the support (including features as complex textures e.g.) or motion characteristics (which can introduce artifacts linked to sudden changes along the motion such as edges breaking, e.g.) see [5]. In the present paper we leave aside statical approaches and focus on analytic and geometric based models.

On the one hand, the goal of analytical approaches is to improve or restore 'infinitesimal' properties according to the time variation of local information around damaged pixels. It can be considered as a planar approach because it uses information about continuity respect to t of planar regions $r(t)$ and their boundaries $b(t) = \partial r(t)$. Usually, analytical approaches are linked to some variant of the gradient $\nabla b(t)$, which is geometrically reinterpreted as the normal flow of the boundary $b(t)$ itself. If the region $r(t)$ is convex, the *normal outward propagation* has good properties: it can be shrinked to a point. However, furthermore usual inconvenient of the gradient flow (as directional field), the normal propagation displays a lot of 'pathologies'. Typical examples are given by different kinds of self–intersections and cusps (linked to normal envelopes or caustics, e.g.) which degrade local computations linked to convolution filters.

Alternately, in the geometric framework one may consider several mathematical tools closely related to the spatial–time geometric structure of the video sequence. Our proposal is based on the very well known Mean Curvature Flow (MCF hereinafter) which incorporates volumetric information about shrinking or expanding phenomena contained in neighbor images of the video mini–sequence. A good recent survey for the MCF of surfaces can be read in [6]. Recall that the MCF is defined as the divergence of the unit normal vector of the surface.

From the Classical Differential Geometry viewpoint, a smooth surface $S \subset \mathbb{R}^3$ is locally characterized at each point $p \in S$ (up to rigid motions) by the Mean Curvature H and the Total Curvature K (Gauss Theorem) which is an intrinsic invariant. They are respectively defined as the arithmetical mean $H := (\kappa_1 + \kappa_2)/2$, and geometric mean $K := \kappa_1\kappa_2$, which are the elementary symmetric functions of the principal curvatures κ_1, and κ_2. In fact κ_1, and κ_2 stand for the eigenvalues of the curvature matrix corresponding to minimal and maximal values for sectional curvatures, i.e. Sections by hyperplanes containing the normal vector $\mathbf{N}_p S$ to S at $p \in S$ (see [7]).

The use of the MCF is not a novelty for propagation models on surfaces; it has been used from long time ago for different purposes, including image and video restoration. In fact to best of our knowledge the original MCF level set equation was proposed in [8], and further generalizations in [9–12]. A classical reference is [13].

Let us remark that video restoration is based on the preservation of the Mean Curvature over the time by non–trivial isometric deformations. The above condition is equivalent to the existence of a conformal metric for which one can compute the evolution of Mean Curvature H and Total Curvature K of a third order ordinary differential equation which allows the construction of an isometric deformation (Codazzi equation [14]) keeping the MCF [15]. Unfortunately, in video sequences one cannot expect so good conditions. Hence, several alternative approaches has been developed from the late nineties. Let us mention the one in [16] for *normal inward direction* with speed equal to the Harmonic Mean Curvature H/K for a surface in the ordinary space. This method provides explicit solutions with a good behavior for revolution surfaces.

Nevertheless its interest linked to normal inward propagation, their applicability to video sequences is limited to a very specific kind of surfaces. In [17] the authors develop an approach based on m–th powers of the Mean Curvature going farther the cases corresponding to convex hypersurfaces converging to a sphere. The strategy consists of using volume–preserving MCF to obtain finer convergence results (towards a sphere of radius $r > 0$), by extending previous results of [18].

The use of the MCF associated to an evolving surface M_t: $\mathbb{R}^2 \mapsto \mathbb{R}^3$, $t \geqslant 0$ displays some inherent troubles. To understand them, let us observe that MCF is defined as the solution of a quasilinear parabolic equation

$$\frac{\partial H(M_t)}{\partial t} = \Delta_{g(t)} H(M_t) + |A|^2 H(M_t), \qquad t \geqslant 0, \tag{1}$$

where A is the second fundamental form of the evolving surface M_t, $\Delta_{g(t)}$ is the classical Laplace–Beltrami operator, and $g(t)$ stands for a given metric tensor. In other words, it is a quasilinear heat equation with a superlinear growth [19] which gives an explosive behavior of MCF in finite time, and forces to a very careful analysis of discrete approaches taking into account the length of the time intervals. Different alternatives have

been proposed in the literature. A first option consists of combining short–time analytic conditions along with the first step of iteration (expressed in numerical terms). Another option consists of obtaining geometric stability conditions for the partial reconstruction, by means of a suitable regularizations. Last ones are discussed more in depth in section 2.

Based on the above considerations our first proposal consists of an integral approach to MCF based on (1). In particular our first approach consists of developing a video restoration model based on a nonlinear and non–local in time MCF formulation arising from the time integration of (1), that is

$$H(M_t) = H(M_0) + \int_0^t \{\Delta_{g(s)} H(M_s) + |A|^2 H(M_s)\}\, \mathrm{d}s, \qquad t \geqslant 0. \tag{2}$$

What is more, we consider a further generalization of (2) in terms of nonlinear convolution equations

$$H(M_t) = H(M_0) + \int_0^t \mathcal{K}(t-s)\{\Delta_{g(s)} H(M_s) + |A|^2 H(M_s)\}\, \mathrm{d}s, \qquad t \geqslant 0. \tag{3}$$

where $\mathcal{K}(t)$ stands for a suitable convolution kernel. This choice is proposed for a twofold reason:

1. The proposed approach allows us to take into account a larger amount of information back in time. In the discrete setting this means that one can take into account more frames back apart from the current and previous ones.
2. The regularization effect of the convolution approach is expected to cut down the singularities associated to the classical approach (1).

The practical results achieved with several video mini–sequences show that the classical MCF (1) (i.e. if one goes back in time one frame), provides little competitive performances. Moreover we have observed that if more than one frame is considered according to (3) instabilities remain, and what is worse, the more frames back we consider, the worse the restoration is. In particular unexpected artifacts come to appear as one goes further back in time. This suggests that geometric features associated to the MCF make unstable the propagation model for large time scales, therefore unsuitable for our purposes.

Our *second contribution* consists of developing an analytical approach, according to the classification above, where geometrical characteristics do not play a key role. Instead, we assume a 'short time coherence'. In other words, we assume that the video sequence displays some kind of local continuity for the intensity at every single pixel around the damaged area; this region is called a 'collar' in Differential Topology. From an analytic viewpoint, this 'timeline continuity' is modeled by means of a classical linear Volterra equation (known as well as linear non–local in time evolution equation), both from a continuous viewpoint

$$M_t = M_0 + \int_0^t \mathcal{K}(t-s) M_s\, \mathrm{d}s, \qquad t > 0, \tag{4}$$

but also from its discrete counterpart, namely in terms of a discrete set of frames as described in section 2.2. Instead of considering geometric properties of the MCF, the convolution kernel $\mathcal{K}(t)$ is expected to regularize instabilities associated to large time scales ocuring in practical situations. Let us remark that $\mathcal{K}(t)$ explicitly arises only in the discrete context where it is computed according to particular features of the sequence. The convolution weights arising in the discrete formulation are computed by means of the Least Squares Method following similar ideas to those appearing in machine Learning methodologies.

The experimental results revealed the good performance of this approach including the stability as we go back in time, that is if one uses the information provided by several previous frames, up to some threshold. Our results show that this approach provides by far better performance that (2), also. Notice that in this approach subsequent frames could be considered as well, but not only previous ones, however this issue keeps out of the scope of the present work. To the best of our knowledge, this approach has not been considered before in the literature, and provides a very low computational cost and effective procedure for video restoration.

Other approaches have been also considered in recent literature for video restoration, let us mention some based on geometrical features of the video sequence as e.g. motion compensation (see e.g. [20]), optical flow (see e.g. [21]), or fractional optical flow (see e.g. [22]) among others. All these approaches are more sophisticated that the one proposed in (4). Hence, they are much more expensive in terms of computational runtime; what is more, their stability has not been always accurately studied. By these reasons, we focus on the two approaches described above in order to show their performance, and mainly in (4) owing to its analytic nature, as more efficient alternative to the geometric one. Also, we show that the number of frames to take into account in each mini–sequence restoration must be somehow limited. Otherwise, undesired artifacts in the restored frame begin to appear.

Finally, our research does not focus on an automatic location of damaged/lost pixels since this issue deserves his own development. Instead, to localize damaged/lost pixels in our experiments we make a partial use of existing algorithms (see e.g. [3], Ch. 4, or [23]).

According to the above remarks, the contents of the article is as follows: section 2 is devoted to give an accurate formulation of MCF based approaches, and its further discussion. In this section we also describe how convolutions weights involved in the non–local approaches are in practice computed. In section 3 we describe in detail our second proposal based on a linear convolution equation. The paper concludes with section 4 where a deep result assessment of our proposals is carried out by means of a number of suitable practical experiments.

2. Mean curvature flow

Let us fix the notation to be used throughout the paper. In fact in the continuous setting we denote by $\{M_t\}_{0\leqslant t\leqslant T}$ a one–parameter set of surfaces in $\mathbb{R}^3$ corresponding to a video mini–sequence. Their implicit representation can be ideally written as the level set of maps, $\psi(\cdot,t)\colon \mathbb{R}^3 \to \mathbb{R}, 0 \leqslant t \leqslant T$, that is

$$M_t := \{\mathbf{x} = (x, y, z)^T \in \mathbb{R}^3 \colon \psi(\mathbf{x}, t) = 0\}, \qquad 0 \leqslant t \leqslant T. \tag{5}$$

Alternately, one may assume that surfaces M_t admit an explicit representation, that is there exists $u(\cdot,\cdot,t)\colon \mathbb{R}^2 \to \mathbb{R}, 0 \leqslant t \leqslant T$, so that

$$M_t := \{(x, y, u(x, y, t)) \colon x, y \in \Omega\}, \qquad 0 \leqslant t \leqslant T. \tag{6}$$

Notice that in that case M_t stands for the level sets of $\psi(x, y, z, t) = z - u(x, y, t), 0 \leqslant t \leqslant T$, or equivalently for the solutions of the equation $z = u(x, y, t), 0 \leqslant t \leqslant T$, where for the sake of the simplicity of the notation we avoid the dependence on t of z.

The representation (6) is suitable for the presentation of our results this is why now and hereafter we merely focus on such representation.

2.1. Classical mean curvatura flow

According to the explicit representation (6) of a surface the standard Curvature Flow for the family of surfaces $\{M_t\}_{0\leqslant t\leqslant T}$ is defined as

$$\mathcal{H}(M_t) = \mathcal{H}(u(\mathbf{x}, t)) = \frac{1}{2}\,\mathrm{div}\left(\frac{\nabla u(\mathbf{x}, t)}{\|\nabla u(\mathbf{x}, t)\|}\right), \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} = (x, y) \in \Omega, \tag{7}$$

where $\Omega \subset \mathbb{R}^2$ stands for the spatial domain, and the operators *div*, and ∇ stand for the divergence and gradient operators, respectively, on plane variables (x, y). Therefore the differential formulation of the standard MCF reads

$$\frac{\partial u}{\partial t}(\mathbf{x}, t) = \mathcal{H}(u(\mathbf{x}, t)) = \frac{1}{2}\,\mathrm{div}\left(\frac{\nabla u(\mathbf{x}, t)}{\|\nabla u(\mathbf{x}, t)\|}\right), \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \Omega, \tag{8}$$

with the initial and boundary conditions

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \quad \mathbf{x} \in \Omega, \qquad \text{and} \qquad \frac{\partial u}{\partial n}(\mathbf{x}, t) = 0, \quad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \partial\Omega, \tag{9}$$

respectively. In (9) u_0 stands for the initial data of the problem, and in p ur framework to the first frame considered in the restoration process. Moreover $\partial\Omega$ stands for the boundary of Ω. Therefore (8)–(9) gives rise to the parabolic initial and boundary value problem to be solved.

According to the remarks appearing in the introduction, there can appear degenerate solutions of (8). This occurs when the denominator in (8) vanishes, that is if $\nabla u = \vec{0}$. Therefore a suitable regularization may be seen as a possible solution. In this regard different regularizations have been proposed in the literature, including the regularized mean curvature $\mathcal{R}$ instead of $\mathcal{H}$ proposed in [24] and defined as

$$\mathcal{R}(u) = \int_\Omega |\mathcal{H}(u)|^q \, \mathrm{d}\Omega, \tag{10}$$

where $q > 0$, typically $q = 1$ or $q = 2$, or e.g. the one proposed by Evans–Spruck in [25]. In the present paper we focus in the last one which, according to the explicit representation of surfaces, can be writen as

$$\mathcal{H}(M_t) = \mathcal{H}(u(\mathbf{x}, t)) = D\,\mathrm{div}\left(\frac{\nabla u(\mathbf{x}, t)}{\sqrt{\epsilon + \|\nabla u(\mathbf{x}, t)\|^2}}\right) \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \Omega, \tag{11}$$

where $D > 0$ stands for the diffusion coefficient, and $\epsilon > 0$ for the regularizing parameter, thought it is expected to be small. Therefore the parabolic problem associated to (11) reads as

$$\frac{\partial u}{\partial t}(\mathbf{x},\, t) = \mathcal{H}(u(\mathbf{x},\, t)) = D\,\mathrm{div}\left(\frac{\nabla u(\mathbf{x},\, t)}{\sqrt{\epsilon + \|\nabla u(\mathbf{x},\, t)\|^2}}\right), \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \Omega, \tag{12}$$

along with the initial and boundary conditions (9). The nonlinear parabolic problem (9) and (12) turns out to be well–posed [25].

A more general approach to MCF (see [26]) is written as

$$\frac{\partial u}{\partial t}(\mathbf{x},\, t) = \mathcal{H}(u(\mathbf{x},\, t)), \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \Omega, \tag{13}$$

where $\mathcal{H}$ stands for

$$\mathcal{H}(u(\mathbf{x},\, t)) = g(\|\nabla u(\mathbf{x},\, t)\|)\,\mathrm{div}\left(\frac{\nabla u(\mathbf{x},\, t)}{f(\|\nabla u(\mathbf{x},\, t)\|)}\right), \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \Omega, \tag{14}$$

for a suitable choice of f and g. In particular, MCF (12) consists of taking $f(x) = x$ and $g(x) = D$ (both of them before regularization).

Note that the equation (12) may be explicitly written as

$$\frac{\partial u}{\partial t}(\mathbf{x},\, t) = D\frac{(\epsilon + u_x^2)u_{yy} - 2u_x u_y u_{xy} + (\epsilon + u_y^2)u_{xx}}{(\epsilon + u_x^2 + u_y^2)^{3/2}}, \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \Omega. \tag{15}$$

In regard to the numerical solution of MCF models, let us mention two recent works, the first one involving the finite volume discretization of K Mikula *et al* [26], and the second one of Ch. Lubich *et al* based on the finite element method [27, 28]. However since the nature of our approach is originally discrete, we do not longer pay attention in this paper to the analysis of its numerical solutions.

2.2. Non–local in time mean curvature flow

The MCF process starts with an original surface corresponding to the initial data of the problem (9) and (12). Afterward, solutions to the evolution problem (that is as the time scale evolves) provide new surfaces corresponding to images expected to be a smoothing/filtering/re–construction of the original one.

Our proposed non–local approach consists of considering each video mini–sequence as a set of surfaces $\{M_t\}_{0 \leqslant t \leqslant T}$ evolving over time, and whose structure depends in some manner on the memory of the process further than merely the current and previous surfaces, at least short back in time. This idea comes out from the re–formulation of (12) in integral form simply by integrating it in time. This fact allows us to write (12) as follows

$$u(\mathbf{x},\, t) = u_0(\mathbf{x}) + \int_0^t \mathcal{H}(u(\mathbf{x},\, s))\,\mathrm{d}s, \qquad \mathbf{x} \in \Omega, \quad 0 \leqslant t \leqslant T, \tag{16}$$

where $\mathcal{H}(u(\cdot,t))$ is defined in (11), with boundary conditions as in (9). Note that in (16) the initial condition is explicitly given by $u_0(\mathbf{x})$.

Typical artifacts to be restored are linked to lost pixels/blotches/patches/occlusions, e.g. In this framework of video mini–sequence, we take into account the *memory* of the process, that is in the discrete setting a number of previous frames of the sequence, furthermore the previous one as with (9) and (12). This argument applies only within a limited time back. In other words, this idea works only if we assume 'some time coherence' at least in the pixels belonging to a small neighborhood around the damaged ones.

Our first contribution introduces a more general approach beyond (16), always within the assumption of certain time coherence. In fact, we introduce a convolution kernel in (16) which enable us to handle more accurately the memory of the process according to the features of the sequence. Therefore, the mathematical model we propose writes as the non–linear Volterra equation

$$u(\mathbf{x},\, t) = u_0(\mathbf{x}) + \int_0^t \mathcal{K}(t - s)\mathcal{H}(u(\mathbf{x},\, s))\,\mathrm{d}s, \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \Omega, \tag{17}$$

where in a classical context $\mathcal{K}(t)$ is *a priori* known. More general formulation of non–linear Volterra equations might be considered however these are out of the scope of the present paper.

As we have advanced in the introduction the convolution kernel $\mathcal{K}(t)$ does not play any role in the continuous setting due to the discrete nature of our approach. In the next section, we give an accurate description about how the discrete convolution kernel can be computed in practice according to the features of each single sequence around the damaged area to be restored. This explains why very weak conditions may be required on $\mathcal{K}(t)$ in the continuous setting. So the well–posedness of (17) may be simply derived from the well–posed conditions for (12).

Since the discrete framework is the one where we will work on, we now show the discretization of (17), first in time, then in the spatial variables. Firstly, for the time discretization we set $h > 0$, $t_j = jh$, $j = 0, 1, \ldots, N$ ($T = Nh$), (17) becomes the following time discrete non–linear Volterra equation (also known as time discrete

convolution equation)

$$u_n(\mathbf{x}) = u_0(\mathbf{x}) + \sum_{j=0}^{n-1} \mathcal{K}_{n-j}\mathcal{H}(u_j(\mathbf{x})), \qquad 1 \leqslant n \leqslant N, \quad \mathbf{x} \in \Omega, \tag{18}$$

where $u_j(\mathbf{x})$ stands for the approximation to $u(\mathbf{x}, t_j)$, even though in our framework it corresponds to the j–th frame of the mini–sequence. Since the convergence of the numerical scheme is not a matter in this paper, we can set $h = 1$ as the unit time without loss of generality. Thus, the time step length of the discretization is neither a matter. On the other hand, the convolution weights $\mathcal{K}_j$ do not necessarily correspond to $\mathcal{K}(t_j)$. Actually, they arise from a Machine Learning based procedure as we describe in the next section.

The fully discretization of (18) is reached by replacing in $\mathcal{H}$ the spatial continuous derivatives by a convenient finite differences scheme. Once again since the stability of the numerical discretization does not matter in this paper, we adopt a classical second order difference scheme over a uniform spatial mesh of length 1 both in x and y coordinate axis. In particular if we consider the spatial mesh $M := \{(p, q)\colon 0 \leqslant p \leqslant P, 0 \leqslant q \leqslant Q\}$, $P,\ Q \in \mathbb{Z}^+$, then the fully discretization of (18) gives rise, for every single time step $1 \leqslant n \leqslant N$, to the following system of fully discrete non–linear Volterra equations

$$\begin{cases} u_n^{0,0} &= u_0^{0,0} + \sum_{j=0}^{n-1} \mathcal{K}_{n-j}\mathcal{H}_j^{0,0} \\ u_n^{1,0} &= u_0^{1,0} + \sum_{j=0}^{n-1} \mathcal{K}_{n-j}\mathcal{H}_j^{1,0} \\ \vdots & \quad \vdots \\ u_n^{p,q} &= u_0^{p,q} + \sum_{j=0}^{n-1} \mathcal{K}_{n-j}\mathcal{H}_j^{p,q} \\ \vdots & \quad \vdots \end{cases} \qquad (p,\, q) \in M, \tag{19}$$

where

$$\begin{aligned} \mathcal{H}_n^{p,q} := & \frac{D}{(\epsilon + (u_n^{p+1,q} - u_n^{p,q})^2 + (u_n^{p,q+1} - u_n^{p,q})^2)^{3/2}} \cdot \\ & \{(\epsilon + (u_n^{p+1,q} - u_n^{p,q})^2)(u_n^{p,q+1} - 2u_n^{p,q} + u_n^{p,q-1}) \\ & -2\,(u_n^{p+1,q} - u_n^{p,q}) \cdot (u_n^{p,q+1} - u_n^{p,q}) \cdot (u_n^{p+1,q+1} - u_n^{p,q+1} - u_n^{p+1,q} + u_n^{p,q}) \\ & +(\epsilon + (u_n^{p,q+1} - u_n^{p,q})^2)(u_n^{p+1,q} - 2u_n^{p,q} + u_n^{p-1,q})\}. \end{aligned} \tag{20}$$

The algorithm (20) is fully stated if combined with the discrete initial condition

$$u_0^{p,q} = u_0(p,\, q), \quad 0 \leqslant p \leqslant P,\ 0 \leqslant q \leqslant Q, \tag{21}$$

and the discrete boundary conditions (extending the domain by reflection as usual)

$$u_j^{P-1,q} = u_j^{P+1,q}, \quad u_j^{p,Q-1} = u_j^{p,Q+1}, \quad (p,\, q) \in M. \tag{22}$$

Notice that in a numerical context $u_j^{p,q}$ stands for the approximation of $u(p, q, j)$, however, in the present context it represents the gray value of pixel (p,q) in the frame j.

2.3. Computation of convolution weights

The computation of convolution weights is worthy of the current section, where we explain how they can be in practice computed. We are interested in the convolution weights in (19) rather than the convolution kernel $\mathcal{K}(t)$ itself, which at first are not *a priori* known. The underlying idea we propose here is based on Machine Learning methodologies and follows the following steps:

1. Locate the set of damaged pixels $L = \{p_1, p_2, \ldots, p_{M_l}\}$ in a given frame F_0. These pixels belong to the patch to be restored. For the sake of the simplicity of the notation we avoid the Euclidean notation that is the explicit components $p_j = (p_{j,1}, p_{j,2})$. Therefore the value $u_{F_0}^{p_j}$, $1 \leqslant j \leqslant M_l$ corresponds to the grey level of damaged pixel p_j in the frame F_0.
2. Assume that the patch can be surrounded by a set of M_r undamaged pixels denoted by $R = \{q_1, q_2, \ldots, q_{M_r}\}$. Once again we use an abbreviated notation for these pixels.
3. Assume also that the set of pixels $L \cup R$ in $N_R > 0$ previous frames are not damaged.
4. Now if we denote F_{-n} the n–th frame back from the current one F_0, then we use the set of pixels R over the frames F_0 and F_{-1}, $F_{-2}, \ldots, F_{-N_R}$ to learn the weights in (19). Here the main idea of the paper applies, that is we

apply here the *local coherence*, which means that not damaged pixels satisfy a local equation of type (19) where the weights have to be determined. In particular this leads to solve a $M_r \times N_R$ system of equations arising from the set of pixels R, with N_R unknowns, i.e. the set of weights $\{\mathcal{K}_j\}_{j=1}^{N_R}$, which reads

$$\mathbf{u}_{F_0} = \mathbf{u}_0 + \sum_{j=1}^{N_R} \mathcal{K}_j \mathcal{H}_{F_{-j}}, \tag{23}$$

where

$$\mathbf{u}_{F_0} = \begin{bmatrix} u_{F_0}^{q_1} \\ u_{F_0}^{q_2} \\ \vdots \\ u_{F_0}^{q_{M_r}} \end{bmatrix}, \ \mathbf{u}_0 = \begin{bmatrix} u_{F_{-N_R}}^{q_1} \\ u_{F_{-N_R}}^{q_2} \\ \vdots \\ u_{F_{-N_R}}^{q_{M_r}} \end{bmatrix}, \ \mathcal{H}_{F_{-j}} = \begin{bmatrix} \mathcal{H}_{F_{-j}}^{q_1} \\ \mathcal{H}_{F_{-j}}^{q_2} \\ \vdots \\ \mathcal{H}_{F_{-j}}^{q_{M_r}} \end{bmatrix}, \qquad j = 1,\ldots, N_R. \tag{24}$$

Notice that in this context the initial value $\mathbf{u}_0$ stands for the set of values of pixels of the first image we take into account in the discrete evolution process corresponding to the patch we are working with.

5. Since in our framework $N_R < M_r$ the system of equations (23) gives rise to an overdetermined system and its solution $\{\mathcal{K}_j\}_{j=1}^{N_r}$ is understood in the sense of Least Squares Method. Note that we are allowed to apply the Least Squares Method as shown below since the equations of system (23) are all of them implicit. In fact, that system may be written in matrix form as

$$\mathbf{u}_{F_0} = \mathbf{u}_0 + \mathcal{H}\mathcal{K}, \quad \text{where} \quad \mathcal{H} = \begin{bmatrix} \mathcal{H}_{F_{-1}}^{q_1} & \mathcal{H}_{F_{-2}}^{q_1} & \cdots & \mathcal{H}_{F_{-N_R}}^{q_1} \\ \mathcal{H}_{F_{-1}}^{q_2} & \mathcal{H}_{F_{-2}}^{q_2} & \cdots & \mathcal{H}_{F_{-N_R}}^{q_2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathcal{H}_{F_{-1}}^{q_{M_r}} & \mathcal{H}_{F_{-2}}^{q_{M_r}} & \cdots & \mathcal{H}_{F_{-N_R}}^{q_{M_r}}, \end{bmatrix}, \mathcal{K} = \begin{bmatrix} \mathcal{K}_1 \\ \mathcal{K}_2 \\ \vdots \\ \mathcal{K}_{N_R} \end{bmatrix}. \tag{25}$$

The solution provided by the Least Squares Method for the system (25) comes out by solving the associated normal equations, that is

$$\mathcal{H}^T(\mathbf{u}_{F_0} - \mathbf{u}_0) = \mathcal{H}^T\mathcal{H}\mathcal{K},$$

whose solution $\mathcal{K}$ stands for the wanted set of weights $\{\mathcal{K}_j\}_{j=1}^{N_R}$.

6. The restoration of damaged pixels comes through the weights $\{\mathcal{K}_j\}_{j=1}^{N_R}$ computed above. This means that the same *local coherence* as for the set of pixels R applies to the set of lost pixels L according to (19) as follows

$$\mathbf{v}_{F_0} = \mathbf{v}_0 + \sum_{j=1}^{N_R} \mathcal{K}_j \mathcal{H}_{F_{-j}}, \tag{26}$$

where

$$\mathbf{v}_{F_0} = \begin{bmatrix} u_{F_0}^{p_1} \\ u_{F_0}^{p_2} \\ \vdots \\ u_{F_0}^{p_{M_l}} \end{bmatrix}, \ \mathbf{v}_0 = \begin{bmatrix} u_{F_{-N_R}}^{p_1} \\ u_{F_{-N_R}}^{p_2} \\ \vdots \\ u_{F_{-N_R}}^{p_{M_l}} \end{bmatrix}, \ \mathcal{H}_{F_{-j}} = \begin{bmatrix} \mathcal{H}_{F_{-j}}^{p_1} \\ \mathcal{H}_{F_{-j}}^{p_2} \\ \vdots \\ \mathcal{H}_{F_{-j}}^{p_{M_l}} \end{bmatrix}, \qquad j = 1,\ldots, N_R. \tag{27}$$

7. Therefore the entries of vector $\mathbf{v}_{F_0}$ stands the set of restored pixels.

3. A linear Volterra type based approach

The practical experiments carried out in section 4 show that the behavior of MCF (9) and (12), that is MCF in the classical sense, provides accurate results in the restoration of damaged pixels. However the non–local in time MCF (17), or more precisely in a discrete format (19), does not improve in general such a results. Instead, the algorithm performance worsened if more than one frame back is used in the restoration.

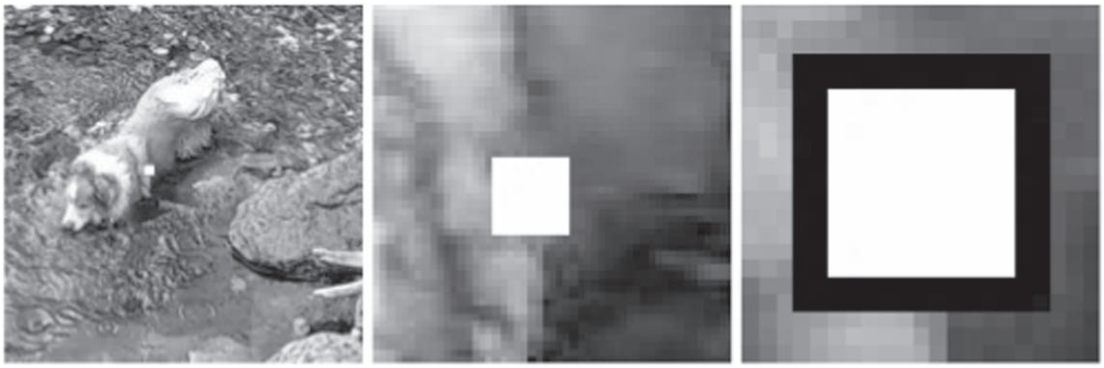

Figure 1. (From the left to the right) Original frame once it has been damaged; Damaged area zoomed; Damaged area zoomed again, and training pixels (in black). Data of the experiment: $\#L = 121$ (lost pixels), $\#R = 104$ (training pixels).

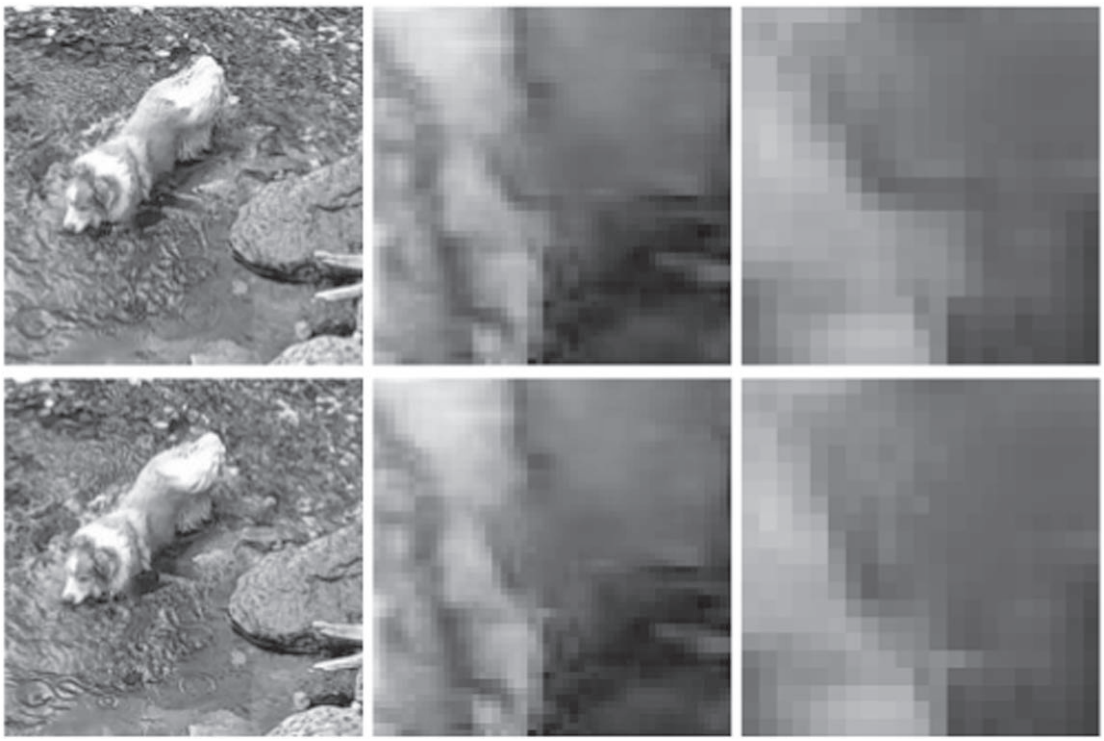

Figure 2. Classical MCF ($N_R = 1$). First row: Original frame; area of interest zoomed; and area of interest zoomed again. Second row: Restored frame; restored area zoomed; and restored area zoomed again.

Our experiments have shown that a simpler approach turned out to be much more accurate. In particular, such approach consists of replacing the operator $\mathcal{H}(u(\cdot,t))$ in (17) by the identity operator $\mathcal{I}(u(\cdot,t))$. This leads to the non–local in time equation

$$u(\mathbf{x},\, t) = u_0(\mathbf{x}) + \int_0^t \mathcal{K}(t-s)u(\mathbf{x},\, s)\, \mathrm{d}s, \qquad 0 \leqslant t \leqslant T, \quad \mathbf{x} \in \Omega, \tag{28}$$

along with the same boundary conditions (9). Again the initial condition $u_0(\mathbf{x})$ appears in (28) explicitly. Moreover a fully discrete formulation of (28), in the same manner as in (17), leads to solve at single time step $0 \leqslant n \leqslant N$, the system of linear equations

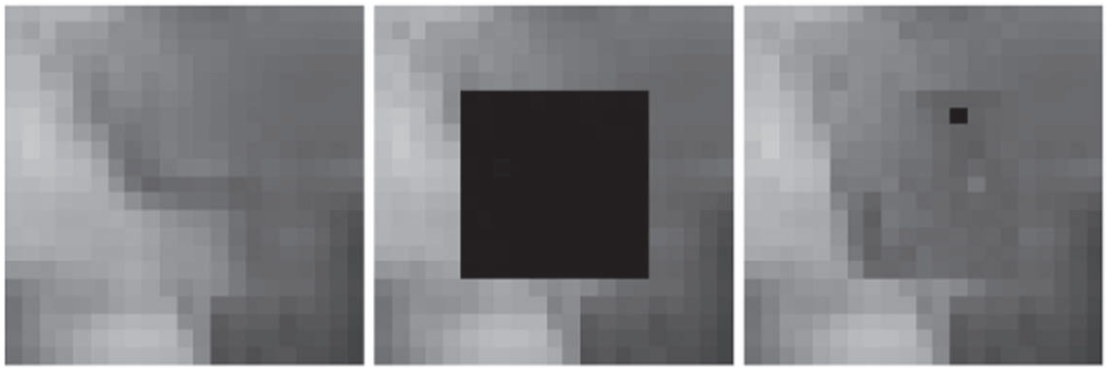

Figure 3. Non–local MCF with $N_R = 2$ (only zoomed frame). Form left to right: Original frame, lost pixels, and restore frame.

Figure 4. From left to the right: Original frame (zoomed); damaged frame (zoomed); and set of removed pixels (in white) surrounded by training pixels (in black). Data of the experiment: $\#L = 102$ (lost pixels) and $\#R = 228$ (training pixels).

Table 1. Non–local MCF (see figures 2 and 3).

N_R	1	2	3	4	5
Error	0.80%	1.21%	1.24%	1.21%	1.79%

$$\begin{cases} u_n^{0,0} &= u_0^{0,0} + \sum_{j=0}^{n-1} \mathcal{K}_{n-j} u_n^{0,0} \\ u_n^{1,0} &= u_0^{1,0} + \sum_{j=0}^{n-1} \mathcal{K}_{n-j} u_n^{1,0} \\ \vdots & \vdots \\ u_n^{p,q} &= u_0^{p,q} + \sum_{j=0}^{n-1} \mathcal{K}_{n-j} u_n^{p,q} \\ \vdots & \vdots \end{cases} \qquad (p,\, q) \in M.$$

According to the notation in section 2.3, we must solve now the $M_r \times N_R$ system of linear equations

$$\mathbf{u}_{F_0} = \mathbf{u}_0 + \sum_{j=1}^{N_R} \mathcal{K}_j \mathbf{u}_{F_{-j}}, \tag{29}$$

Figure 5. From left to the right: Original frame (zoomed); damaged frame; restoration with the classical MCF ($N_R = 1$); and restoration with non–local MCF and $N_R = 2$.

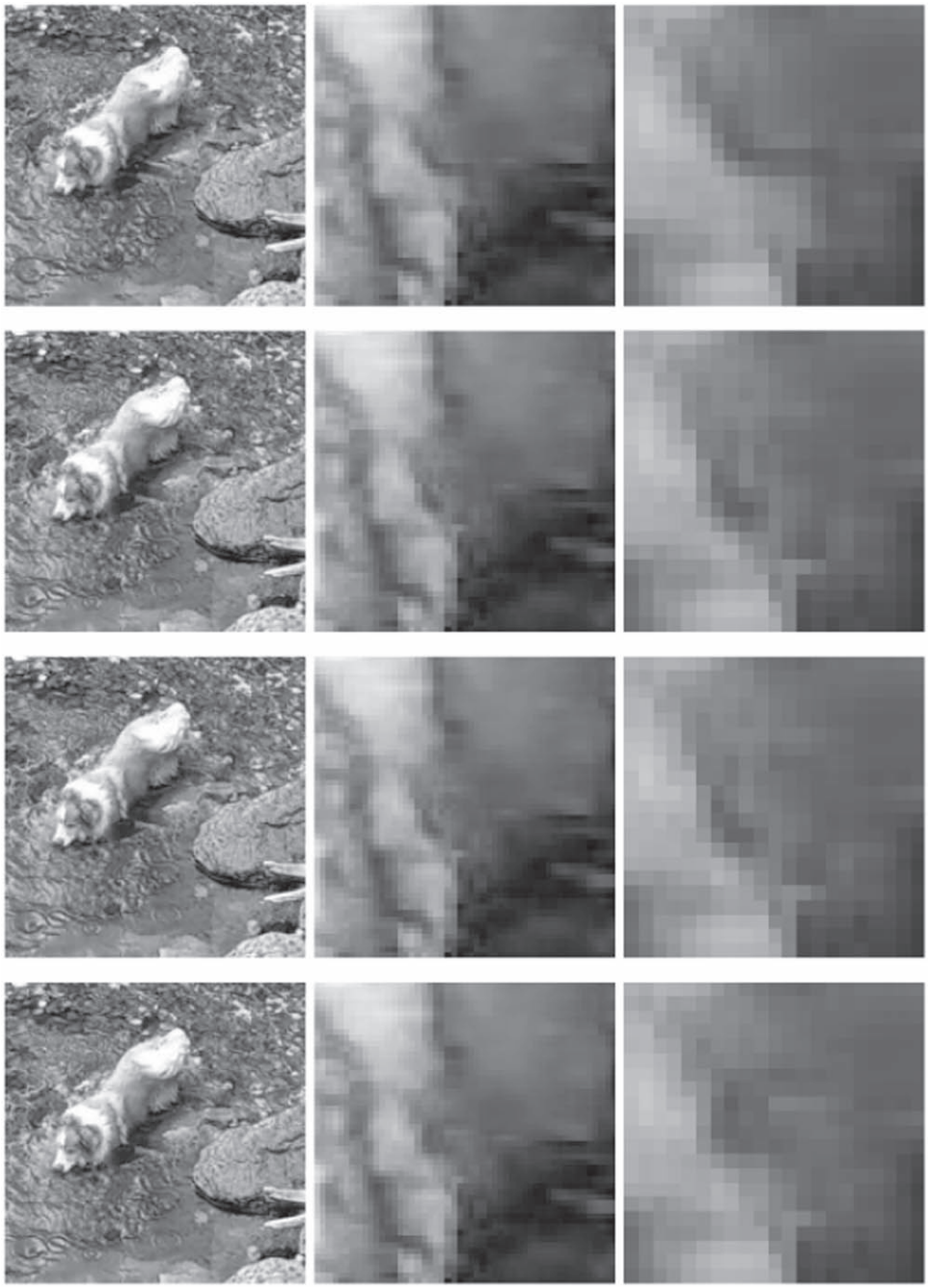

Figure 6. Linear Volterra equations based approach. First row: original frame, zoomed, and re–zoomed. The following three rows: restored whole frame, it zoomed, and re–zoomed; for $N_R = 5$, $N_R = 10$, and $N_R = 30$ each row respectively. Data of the experiment: $\#L = 121$ (lost pixels) and $\#R = 104$ (training pixels).

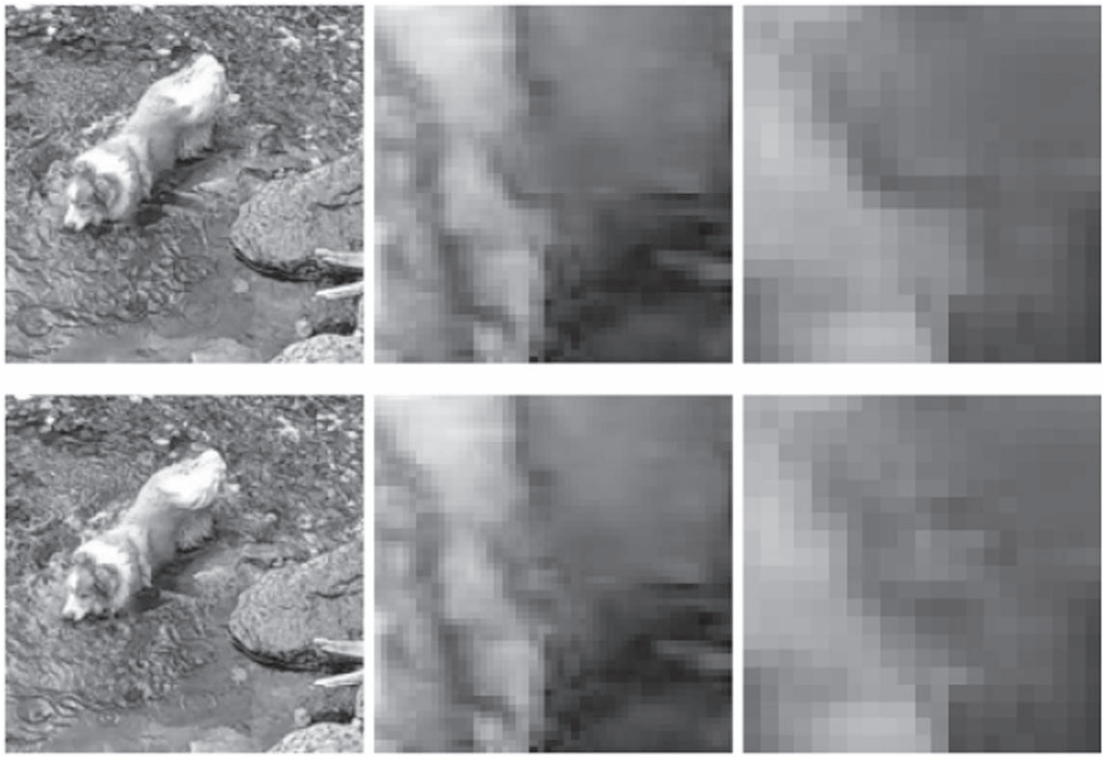

Figure 7. Linear Volterra equations based approach. First row the original frame (first column), and the zoomed and re–zoomed frame (second and third columns). The second row shows the performance with $N_R = 40$ where the results are worse than for $N_R = 30$ (see table 3).

Figure 8. The performance of our approach with $N_R = 5, 10, 20$, and 40.

where

$$\mathbf{u}_0 = \begin{bmatrix} u_{F_{-N_R}}^{q_1} \\ u_{F_{-N_R}}^{q_2} \\ \vdots \\ u_{F_{-N_R}}^{q_{M_r}} \end{bmatrix}, \ \mathbf{u}_{F_{-j}} = \begin{bmatrix} u_{F_{-j}}^{q_1} \\ u_{F_{-j}}^{q_2} \\ \vdots \\ u_{F_{-j}}^{q_{M_r}} \end{bmatrix}, \qquad j = 0, \ldots, N_R. \tag{30}$$

After computing the weights $\{\mathcal{K}_j\}_{j=1}^{N_R}$ by Least Squares Method as we have described in section 2.3, the restored pixels $\mathbf{v}_{F_0}$ are then computed by

$$\mathbf{v}_{F_0} = \mathbf{v}_0 + \sum_{j=1}^{N_R} \mathcal{K}_j \mathbf{u}_{F_{-j}}, \tag{31}$$

Figure 9. From left to the right: Original frame, damaged frame, zoomed damaged part, and training pixels. Data of the experiment: $\#L = 105$ (lost pixels) and $\#R = 120$ (training pixels).

Figure 10. From left to right: MCF with $N_R = 1, 2, 3$ and 4 (only zoomed parts).

Figure 11. From left to right: Linear Volterra equations based approach with $N_R = 10, 20, 30$ and 40 (only zoomed parts).

Table 2. Non–local MCF (see figures 4 and 5).

N_R	1	2	3	4	5
Error	1.08%	1.65%	1.94%	2.52%	2.61%

where

$$\mathbf{v}_0 = \begin{bmatrix} u_{F_{-N_R}}^{p_1} \\ u_{F_{-N_R}}^{p_2} \\ \vdots \\ u_{F_{-N_R}}^{p_{M_l}} \end{bmatrix}, \ \mathbf{u}_{F_{-j}} = \begin{bmatrix} u_{F_{-j}}^{p_1} \\ u_{F_{-j}}^{p_2} \\ \vdots \\ u_{F_{-j}}^{p_{M_l}} \end{bmatrix}, \qquad j = 1,\ldots, N_R. \tag{32}$$

The experiments in section 4 show that going back short in time with (29) noticeably improves the restoration carried out both by the non–local MCF model (19), but also by the classical MCF, that is if $N_R = 1$.

Figure 12. Original full frame; zoomed damaged area; and re–zoomed damaged area.

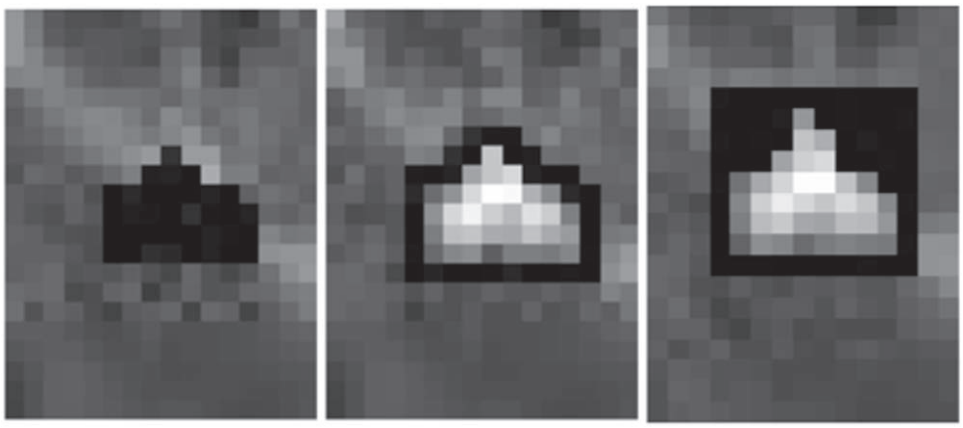

Figure 13. Window of interest corresponding to figure 12: Set of lost pixels (left); and both sets of training pixels (two right).

Table 3. Linear Volterra equations based model (see figures 6 and 7).

N_R	5	10	20	30	40
Error	0.81%	0.80%	0.67%	0.55%	0.70%

Table 4. Linear Volterra equations based model (see figure 8).

N_R	5	10	20	30	40
Error	0.89%	0.84%	0.81%	0.70%	0.72%

4. Numerical experiments

This section is devoted to assess the performance of the proposed approaches, the one based on the MCF, local and non–local in time, and the one based on linear Volterra type equations. The section is divided into three subsections corresponding to three different situations: Non–damaged video sequences which have been artificially damaged in order to accurately measure the quality of performance. Originally damaged video sequences where only a visual perception of the error is allowed; in particular in the second subsection we have considered old movies sequences. Finally, we show how our approach is applied to the restoration of some parts of a sequence which have been occluded by moving objects.

Let us point out that in all experiments shown below the absolute gray scale is preserved. This actually means that original gray scales of each mini sequence and each small window of interest have not been re–scaled, e.g. up

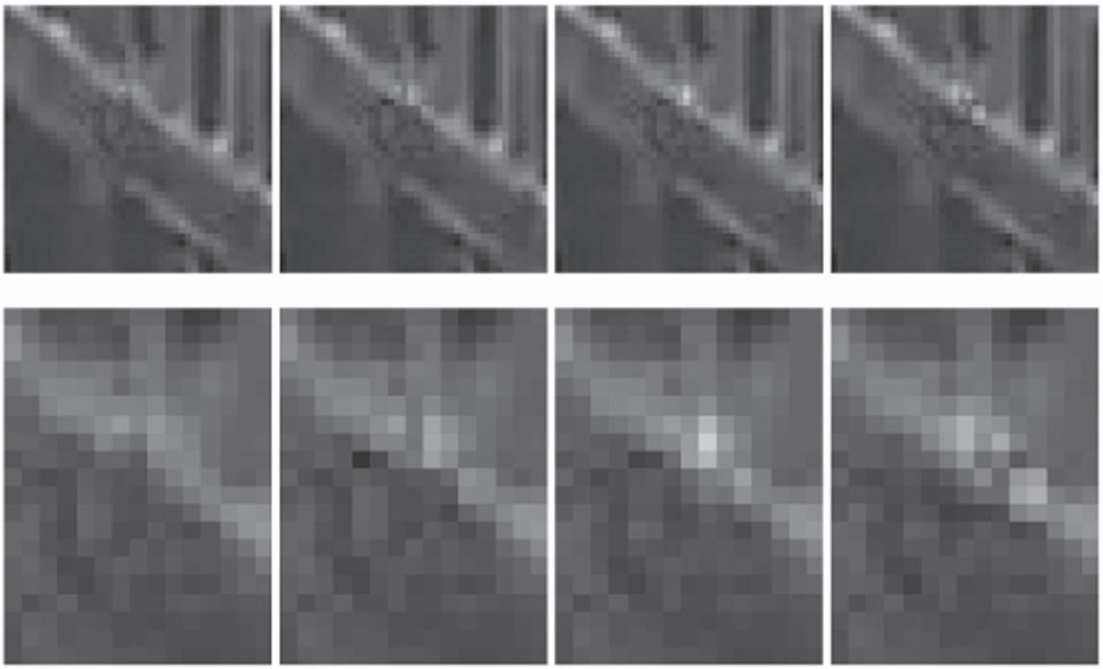

Figure 14. Damaged area (first row) and re–zoomed damaged area (second row) for $N_R = 5, 10, 20$ and 30.

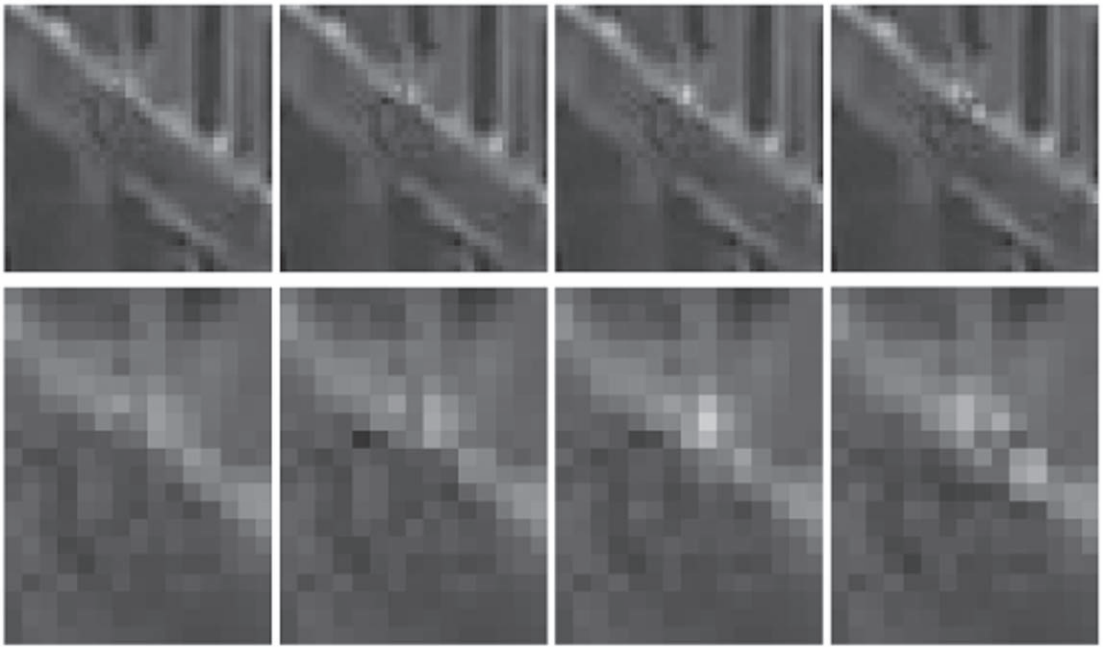

Figure 15. Damaged area (first row) and re–zoomed damaged area (second row) for $N_R = 5, 10, 20$ and 30.

to cover the whole range of values (from 0 to 255). The linearity of our main proposal makes the final results independent on any gray level re–scaling.

Moreover, despite the runtime of both implementations depends on the size of the sets of damage and training pixels, and the number of frame considered, this never has gone beyond a few hundredths of second. Therefore, since in such case the runtime may hardly affected by internal CPU's processes we do not show these data for our experiments.

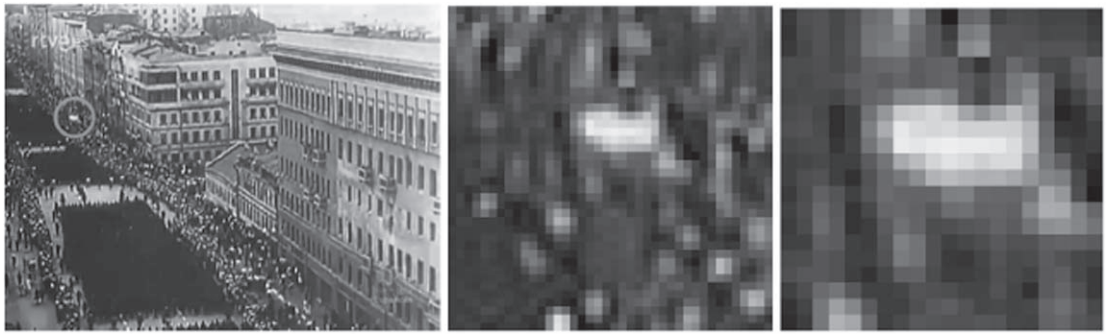

Figure 16. Original full frame; zoomed damaged area; and re–zoomed damaged area.

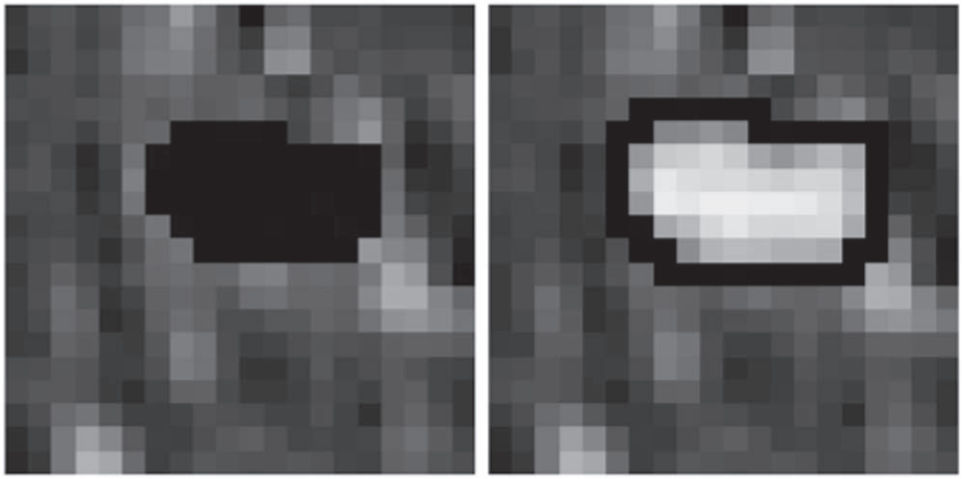

Figure 17. Window of interest corresponding to figure 16: Set of lost pixels; and set of training pixels (both sets in black).

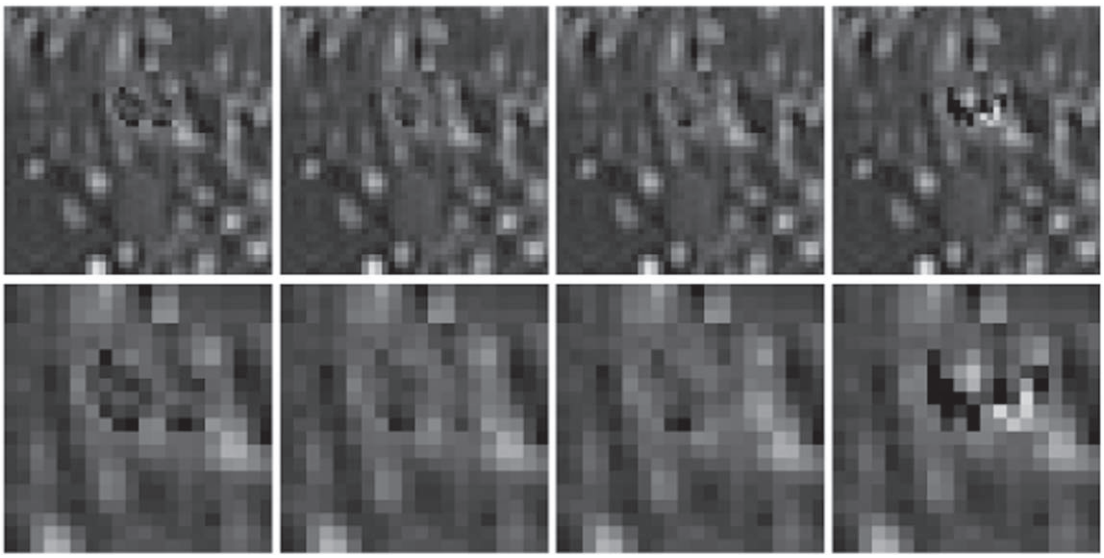

Figure 18. Damaged area (first row) and re–zoomed damaged area (second row) for $N_R = 5, 10, 20$ and 30.

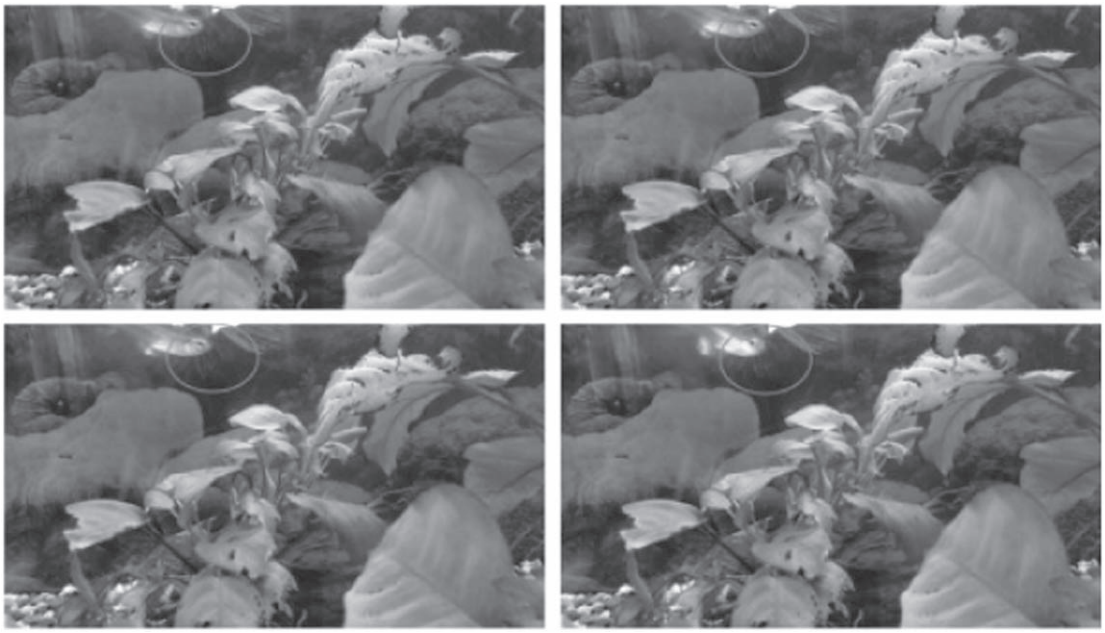

Figure 19. Full original frames with the area of interest circled.

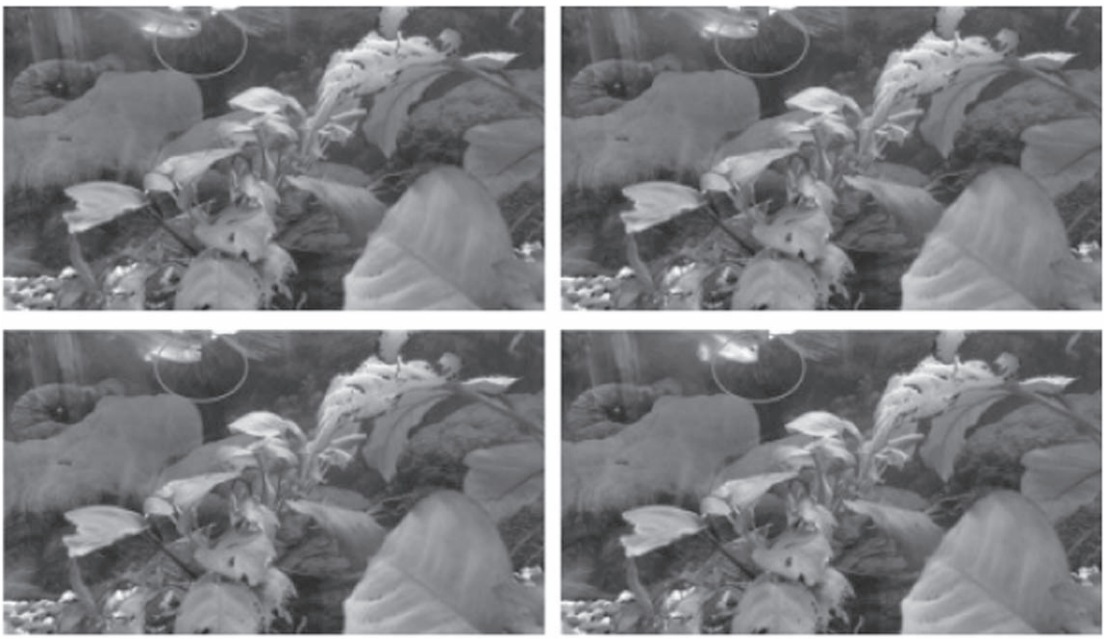

Figure 20. Full original frames with the area of interest circled once it has been recovered the occluded part.

4.1. Test movies

The sequence of frames considered in this subsection has been artificially damaged in different ways. In particular, damaged areas adopt different shapes and sizes in order to get a realistic measure of the performance of the method we propose, and show whether the restoration depends on that or not. In Addition, we show the performance of our methods in some color (RGB format) sequences.

The errors of restorations in the present section have been evaluated according to the following:

- Let $\mathbf{u}_F = [U_F^{p_1}, U_F^{p_2},\ldots,U_F^{p_{M_l}}]^T$ be the vector of original pixels in the current frame F before being damaged. Recall that $L = \{p_1, p_2,\ldots,p_{M_l}\}$ stands for the set of damaged pixels according to the notation in section 2.3, Step 1. Though the inputs of $\mathbf{u}_F$ are ranged from 0 to 255, we normalize the vector to values from 0 to 1 merely by dividing by 255. For the sake of the simplicity of notation we denote the vector again by $\mathbf{u}_F$.

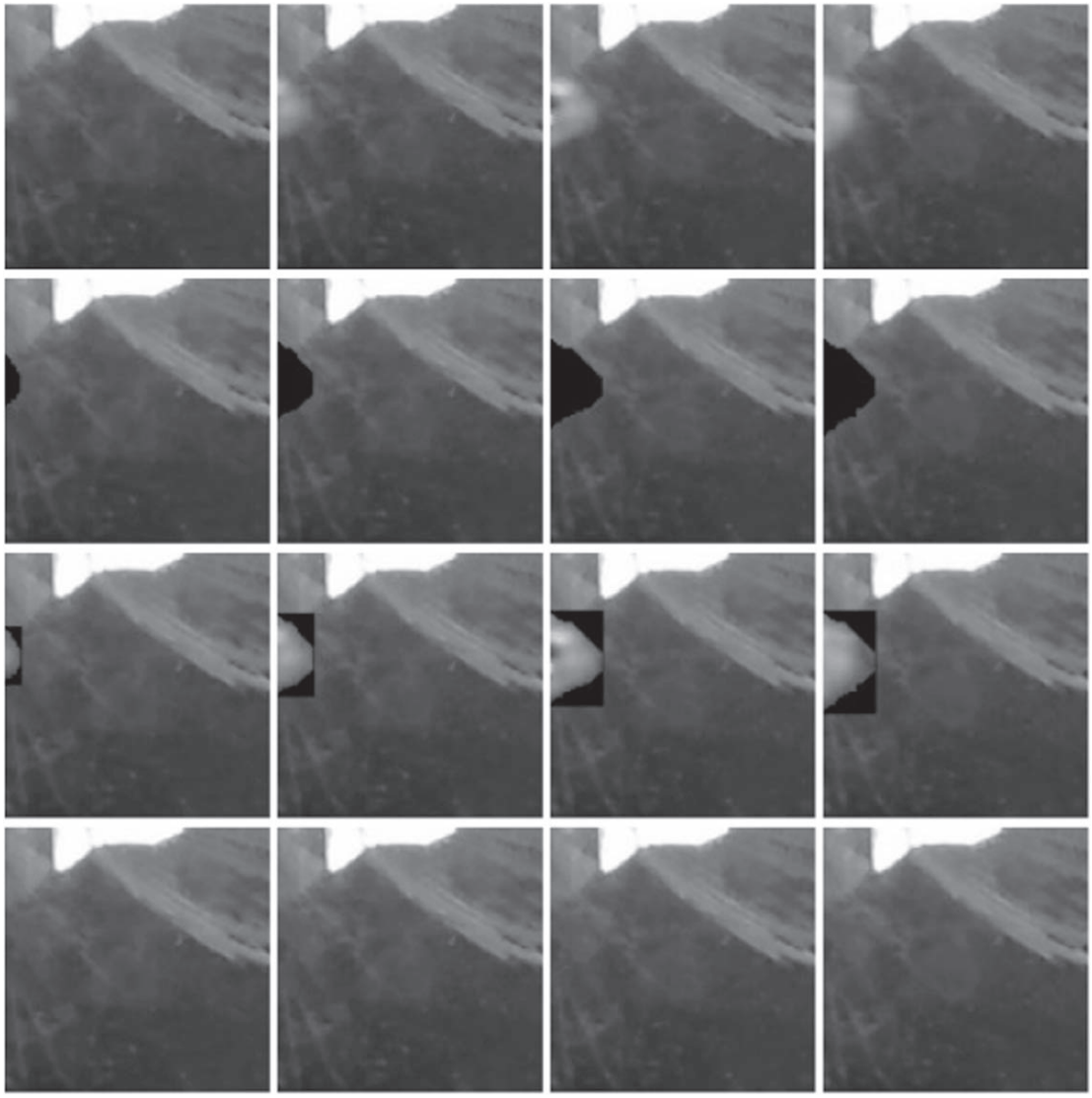

Figure 21. Linear Volterra equations based approach. First row: Original frames zoomed; Second row: Occluding pixels; Third row: Training pixels; Fourth row: Restored frames. All of them performed with $N_R = 5$.

Table 5. Errors in the Red channel for methods (23)–(26), and (29)–(31), and different values of N_R (see figures 10 and 11).

Non–local MCF					
N_R	1	2	3	4	5
Error Red Channel	1.46%	1.69%	1.54%	1.55%	1.53%
	Linear Volterra model				
N_R	10	20	30	40	50
Error Red Channel	1.47%	1.12%	0.66%	0.55%	0.56%

- Let $\mathbf{v}_F = [v_F^{p_1}, v_F^{p_2},\ldots,v_F^{p_{M_I}}]^T$ be the vector of pixels, again in the frame F, after being restored according to Steps 1–6 of section 2.3. We normalize the vector also here to values from 0 to 1, and once again for the sake of the simplicity of notation we denote the vector in the same manner.
- The error is measured by means of the discrete L^2–norm, however since the error is in fact affected by the number of pixels damaged, we relate the error to M_I. In fact the error is computed as follows

$$\text{Error} = \frac{\sqrt{\sum_{j=1}^{M_l}(U_F^{p_j} - v_F^{p_j})^2}}{M_l}. \tag{33}$$

Finally we multiply Error by 100 as to have the error percentage, once again keeping the same notation.

4.1.1. First experiment

Firstly we consider the artificially damaged frame of figure 1. In figure 2 the performance of the classical MCF (19), $N_R = 1$, may be observed where a visual assessment suggests that this method provides very accurate results. However, if one compares those results to the ones provided by non–local MCF (23)–(26) with different values of $N_R > 1$, then it is shown that restoration is not longer improved, in fact the larger the number of frames we consider, the worse the restoration is.

Moreover figure 3 shows the results provided by non–local MCF merely with $N_R = 2$ and certainly the restored image reveals that some artifacts begin to appear, that is restoration is not longer improved if compared to the classical MCF. In addition to the visual assessment table 1 shows how the error for different values of $N_R = 1, 2, \ldots, 5$, grows up as N_R get larger.

In the same frame a different set of pixels is removed, with a different size and shape (see figure 4). The results are neither improved if $N_R > 1$ (see table 2). Once again rare artifacts begin to appear if $N_R > 1$ (see figure 5).

4.1.2. Second experiment

Now we comment on the results of our approach (29)–(31). Once again, we firstly consider the damaged frame in figure 1 where the good performance is observed. In particular figure 6 shows the result of the restoration with $N_R = 5$, 10 and 30 frames back in the sequence. In that case, beyond the visual assessment the errors in table 3 support numerically such a good performance, at least if one goes back up to some point. In particular in table 3 that the error begins to increase for $N_r = 40$, or even at earlier stages (see also figure 7).

In this case we also stress the method by considering the damaged area in figure 4. In that case the performance of the method seems to be not so good, because the shape of the damaged area walks over a more non–homogeneous part of the frame (see figure 8). Nevertheless, our method outperforms the local and non–local MCF based ones. Numerical errors may be seen in table 4. In case of having such so non–homogeneous damaged area one always may split it into several smaller area, and treat them separately.

4.1.3. Third experiment

In the current paragraph we show how both methods (23)–(26), and (29)–(31) behaves if colored sequences in RGB format are considered. Here both methods apply channel–by–channel, that is in the Red, Green, and Blue channel separately. To fix ideas, let us consider the frame and damaged area shown in figure 9. The results of the restoration based on (23)–(26) with $N_R = 1, 2, 3$ and 4 frames back is shown in figure 10, and the results based on (29)–(31) with $N_R = 10, 20, 30$ and 40 frames back in figure 11.

It can be seen that the method (29)–(31) largely outperforms (23)–(26). In fact, table 5 displays only errors in the Red channel for different values of N_R. In this table error is never smaller with MCF based methods (except might be in the first case) than with (29)–(31). Furthermore, while errors with (29)–(31) fall down as N_R increases, with (23)–(26) errors hardly keep constants.

4.2. Restoration of damaged movies

We devote this subsection to show how the method (29)–(31) applies to restore old movies that have been damaged along the archive process, recording, or while their conversion to digital format. In view of performance of both methods discussed above, now we merely focus on (29)–(31). Furthermore we show the resulting restoration in two different sequences of a First World War documentary film.

Firstly consider the frame in figure 12 where the patch to be restored is circled. We here compare the performance of the algorithm with two different sets of training pixels:

- In figure 13 we show the set of lost pixel (left image), the first set of training pixels (middle image), and a second set of training pixels (right image).
- In figures 14 and 15 we show the restored frame with the first and second set of training sets in figure 13 (middle and right images respectively), both for $N_R = 5$, 10, 20, and 30.

The visual assessment suggests that both sets of training pixels provide similar results, and that $N_R = 5$ seems to provided the best performance, so there are not needed a large amount of frames back to provide such good results.

Now consider the frame in figure 16 where the patch to be restored is circled. We here compare the performance of the algorithm with two different sets of training pixels:

- In figure 17 we show the set of lost pixels, and the set of training pixels. In this case we only show the results with one training set.
- In figure 18 we show the restored frame, for $N_R = 5$, 10, 20, and 30, where we must highlight that restoration starts to be worsened for$N_R = 30$ (or smaller) as expected when the number of frames considered grows up after some threshold.

4.3. Restoration of occluded objects

This last subsection is devoted to show that the method (29)–(31) may be also applied to recover areas occluded by moving objects. In particular in this experiment we recover a background consisting of moving plants in a fish tank where a fish comes to appear in the scene.

In figure 19 we show the full frames where we apply the recovery of occluded parts in the area enclosed by a circle, and in figure 20 we show the same (full) frames after the removing is done. In this experiment method (29)–(31) has been applied with $N_R = 5$. Moreover in figure 21 we show the window of interest in the frames considered once we have applied the removing and where it can be precisely observed the occluding object (a fish entering in the scene), the training pixels surrounding the occluding pixels, and the final result of the recovery.

It must be highlighted the good performance of the method also in this context of occluding objets entering in a scene.

5. Conclusions

In this paper we have proposed two new approaches to video restoration based on non–local in time mathematical models. One of them in the framework of non–linear equations and based on the geometric features of time–spatial structure of a video sequence. The second one arose from a linear Volterra equation and it is based on the analytic time regularity (so–called here time coherence) of the video sequence. While the first approach did not improved the results obtained in the local context, the second one provided a very good performance in very different situations, always exploiting the information provided by several previous frames, and within a negligible runtime. This method might give rise to automated real–time video restoration algorithms.

Acknowledgments

The first author was supported by GIR (Research Group) of the University of Valladolid: *Modeling, Theory, and Numerical Analysis in Optimization Problems and Evolution equations.*

On the other hand the authors would like to acknowledge to RTVE (Spanish Public Television) for their documentary repository which has allowed us to carry out an important part of our experimentation.

Moreover the authors are deeply grateful for the comments, corrections, and suggestions of reviewer(s) which have certainly helped to improve the presentation of this work.

Data availability statement

All data that support the findings of this study are included within the article (and any supplementary files).

ORCID iDs

E Cuesta https://orcid.org/0000-0002-7508-6545

References

[1] Kappeler A, Yoo S and Katsaggelos A K 2016 Video super-resolution with convoutional neural networks *IEEE Trans. Comput. Imaging* **2** 109–22

[2] Lagendijk R L, Biemond J, Rares A and Reinders M J T 2009 Ch 4 - video enhancement and restoration *The Esenctial Guide to Video Processing* ed A Bovik (Boston: Academic Press) pp 69–108

[3] Lagendijk R L, van Roosmalen P M B, Biemond J, Rares A and Reinders M J T 2005 3.11 - video enhancement and restoration *Handbook of Image and Video Restoration, Communication, Networking and Multimedia* ed A Bovik II edn (Academic Press) pp 275–VI
[4] Rares A 2004 Archieved film analysis and restoration *PhD Thesis* Delft University of Technology. The Netherlands
[5] Bovik A 2005 *Handbook of Image and Video Processing* II edn (Elsevier) 99–296
[6] Haslhofer R 2021 Lectures on mean curvature flow of surfaces arXiv:2105.10485
[7] Do Carmo M 2017 *Differential Geometry of Curves and Surfaces* II ed (Dover Publications Inc)
[8] Osher S and Sethian J 1988 Fronts propagating with curvature dependent speed: algorithms based on Hamiltonian-Jacobi formulation *J. Comp. Phys.* **79** 12–49
[9] Álvarez L, Guichard F, Lions P L and Morel J M 1993 Axioms and fundamental equations of image processing *Archive for Rat. Mech. Anal.* **123** 200–57
[10] Angentent S B and Gurtin M E 1989 Multiphase thermodynamics with and interfacial structure 2. Evolution of an isothermal interface *Arch. Rat. Mech. Anal.* **108** 323–91
[11] Caselles V, Kimmel R and Sapiro G 1997 Geodesic active contours *Int. J. Comput. Vision* **22** 61–79
[12] Kychenassamy S, Kumar A, Olver P, Tannenbaum A and Yezzi A 1996 Conformal curvature flows: From phase transitions to active vision *Arch. Rat. Mech. Anal.* **134** 275–301
[13] Malladi R and Sethian J A 1995 Image processing via level set curvature flow (image enhancement/image smoothing/geometric heat equation) *Proc. Natl. Acad. Sci. USA (Applied Mathematics)* **92** 7046–50
[14] Aledo J A, Espinar J M and Gálvez J A 2010 The Codazzi equation for surfaces *Adv. Math.* **224** 2511–30
[15] Xiuxiion C and Chia-Kuei P 1989 Deformation of surfaces preserving principal curvatures (LNM 1369.) *Differential Geometry and Topology (Proceedings Tianjin 1986-87)* ed C K Peng, B Jiang and Z Hou (Berlin: Springer) pp 63–70
[16] Daskalopoulos P and Hamilton R 2006 Harmonic mean curvature flow on surfaces of negative Gaussian curvature *Comm. Anal. and Geom.* **14** 907–43
[17] Cabezas-Rivas E and Sinestrari C 2009 Volume-preserving flow by powers of the mth mean curvature *Calc. Var.* **38** 441–69
[18] Huisken G and Polden A 1999 Geometric evolution equations for hypersurfaces *Calculus of Variations and Geometric Evolution* (Berlin, Heidelberg: Springer) (*Lecture Notes in Mathematics*) vol 1713 pp 45–84
[19] Zhu X-P 2002 Lectures on mean curvature flows *Studies in Advanced Mathematics. Amer. Math. Soc. and Inter.* vol 32 (Press of Boston)
[20] Bao W, Lai W-S, Zhang X, Gao Z and Yang M-H 2019 MEMC-Net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement *IEEE Transactions on Pattern Analysis and Machine Intelligence*
[21] Werlberger M, Pock Th, Unger M and Bischof H 2011 Optical flow guided TV-L1 video interpolation and restoration *Conference: Proceedings of the VIII International Conference on Energy Minimization Methods in Computer Vision and Pattern Recognition*
[22] González-Accuña R G, Dávila A and Gutiérrez-Vega C 2019 Optical flow of non-integer order in particle image velocimetry *Signal Process.* **155** 317–22
[23] Khriji L, Meribout M and Gabbouj M 2009 Detection and removal of video defects using rational-based techniques *Adv. Eng. Soft.* **36** 487–95
[24] Gong Y and Goksel O 2019 Weighted mean curvature *Signal Process* 329–39
[25] Evans L C and Spruck J 1991 Motion of level sets by mean curvature I *J. Diff. Geom.* **33** 635–81
[26] Eymar R, Hanlovicova A and Mikula K 2009 Study of a volume scheme for the regularized mean curvature flow level set equation *HAL Id: hal-00407573* http://hal.archives–ouvertes.fr/hal–00407573v2
[27] Kovács B, Li B and Lubich Ch 2019 A convergent evolving finite element algorithm for mean curvature flow of closed surfaces *Numer. Math.* **143** 797–853
[28] Kovács B, Li B and Lubich Ch 2020 A convergent algorithm for forced mean curvature flow driven by diffusion on the surface *Interfaces Free Bound.* **22** 443–64