MOVIE DENOISING BY AVERAGE OF WARPED LINES

By

Marcelo Bertalmío
Vicent Caselles
and
Álvaro Pardo

IMA Preprint Series # 2134

(September 2006)

INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS
UNIVERSITY OF MINNESOTA
400 Lind Hall
207 Church Street S.E.
Minneapolis, Minnesota 55455–0436
Phone: 612/624-6066   Fax: 612/626-7370
URL: http://www.ima.umn.edu

Movie Denoising by Average of Warped Lines

Marcelo Bertalmío, Vicent Caselles and Álvaro Pardo*

Abstract

Here we present an efficient method for movie denoising that does not require any motion estimation. The method is based in the well known fact that averaging several realizations of a random variable reduces the variance. For each pixel to be denoised we look for close similar samples along the level surface passing through it. With these similar samples we estimate the denoised pixel. The method to find close similar samples is done via warping lines in spatio-temporal neighborhoods. For that end we present an algorithm based on a method for epipolar line matching in stereo pairs which has complexity $O(N)$ where N is the number of columns in the image. In this way the algorithm is computationally efficient. Furthermore, we show that the presented method is unsupervised and can deal with different types of noise while respecting the visual details on the movie frames.

I. Introduction

Image denoising is one of the most studied problems in the image processing community. A complete review of all existing methods can easily fill several volumes and is completely out of the scope of this work. Here we are going to content ourselves with an overview of the most related techniques to our work of image sequence denoising.

Within the area of image sequence denoising we can distinguish different cases regarding the source material, the specific type of noise and its application. In our case the motivation to study this problem is the restoration of old films. Nevertheless, the ideas here presented can be applied to other types of image sequences: biological (3D), ultrasound, infrared, compressed (with coding noise), etc.

Before going into details of image sequence denoising we are going to discuss three key issues. First, it is important to consider the different types of noise that can be present in the image sequence. Noise in image sequences can be produced during acquisition by the sensor, due to errors during its transmission, by coding noise, etc. In the case of old films the noise comes from dirt in the film or sometimes the intrinsic film grain is considered as noise to be removed. It is well known that this film grain is signal dependant [1] and therefore spatio-temporal variant. Therefore, it is important to develop methods that can deal with different types of noise.

The second issue is the degradation of the original content of the sequence: we must respect as much as possible the original content of the sequence (details, texture, motion, etc) without introducing artifacts during the denoising process. This is the usual trade off between noise removal and signal degradation. For example, some denoising methods introduce staircasing effects [2], [3] or false contours[1]. The goal is to devise a denoising scheme that can guarantee good denoising capabilities while providing good visual quality. At the end of the day, it may be better to leave some noise instead of removing it together with important elements of the original sequence. In addition to the unpleasant visual distortions that can affect the original content, the degradation of the original content may also affect further processing steps as: segmentation, motion estimation, compression, etc.

Finally, the last issue is the computational complexity of the method and its number of parameters. Due to the enormous amount of data present in image sequences the proposed schemes must be automatic, without a large number of parameters, and computationally lightweight. Although it is not necessary to have a real time method, the method must provide results in a reasonable time in order to allow an interactive process with the user.

Although we can apply existing static image denoising methods to the case of image sequences[2] (intra-frame methods), we can do better including temporal information (inter-frame methods) (see [4]). This temporal information is crucial since our perception is very sensitive to temporal distortions like edge displacement: the disregard

Marcelo Bertalmío and Vicent Caselles are with Departament de Tecnologia, Universitat Pompeu Fabra, Barcelona, Spain (e-mail: {marcelo.bertalmio, vicent.caselles}@upf.edu).

Álvaro Pardo is with DIE, Universidad Catlica del Urguay and on leave from IIE & IMERL, Universidad de la República, Montevideo, Uruguay (e-mail: apardo@ucu.edu.uy).

[1] A similar effect appears in compressed material due to the coarse quantization.

[2] For now on, we will concentrate in temporal sequences leaving out 3D images. However, several of the comments and ideas here presented can be applied to the latter case.

of temporal information may lead to temporal inconsistencies in the result. Filters which take into account the 3D image support can be classified into motion adaptive filters and motion compensated filters [4]. Motion adaptive filters take into account the dynamic character of the sequence but do not compute the optical flow. They are based on averaging pixels of different frames trying to avoid the blurring effect where motion occurs, they are the temporal counterpart of edge preserving spatial filters in that temporal edges are related to motion. Examples include different types of adaptive median filters and order statistic filters [5], [6] or recursive filters [7], [8] (see [4]). Motion compensated filters are based on the assumption that the variation of the pixel gray level over a motion trajectory is mainly due to noise and, thus, averaging these values should give a good estimate of the true pixel value; they produce high quality results. The motion compensated spatio-temporal LMMSE was proposed by [9] and studied in [10] (these filters are an extension of spatial LMMSE filters introduced in [11], [12]). In [13] the authors introduced an Adaptive Weighted Algorithm (AWA) which can be interpreted as a motion compensated neighborhood filter. We refer to [4] for a more detailed account of these methods.

Unfortunately, motion estimation is an ill-posed problem (that needs extra conditions in order to be solved) and its estimation is not straight forward in the case of noisy sequences. To overcome the problem of motion estimation, Buades et. al., based on their previous work [14], presented a method (the NLM method) for image sequence denoising that does not need motion estimation [15]. Starting from the idea that averaging several independent realizations of the same random variable reduces noise, they present a method for image denoising that considers a weighted average of similar samples. Similar samples (pixels) are found comparing their neighborhoods: two pixels with similar neighborhoods are said to be similar. In section III we present this method in detail. A similar method was presented in [16] where the spatio-temporal neighborhoods are locally adapted.

In both cases the basic idea is to find similar samples along the sequence to filter out the noise. An important issue is the unsupervised nature of the proposal. Similar samples are automatically found within the image sequence without any training step. This approach can be rooted to the pioneering work of Efros-Leung [17]. The main feature of these methods is that they may look for similar neighborhoods all over the image (or sequence)[3]. In this way they find many similar samples for the denoising procedure. For a detailed review of neighboring filters and PDE-based methods we refer the reader to [14]. In [1] the authors extend their work on Field of Experts [18] in order to deal with grain noise in archival film. They develop a model of grain noise that is used as a prior in the Field of Experts framework.

The main purpose of this paper is to propose an efficient method for denoising digital image sequences. The main idea behind our proposal is similar to the one used in [15] and [16]. For each pixel we look for a set of similar samples to be used in the filtering step: we estimate the nearby points on the level surface passing trough it, we consider them as realizations of the same random variable and we take an average of them. For that purpose, we present an efficient method to find similar samples via warping lines in spatio-temporal neighborhoods. Our main concerns are: the computational cost of our algorithm, its unsupervised nature, its capabilities to automatically deal with different kinds of noise and its possibilities to respect the visual details on the image sequence. As we will see, the proposed method obtains good denoising results while outperforming the computational complexity of the methods proposed in [15], [16].

The plan of the paper is as follows. In Section II we give a direct introduction to the movie denoising method proposed in this paper. In Section III we discuss the basic properties of the method. Section IV describes the numerical implementation and Section V contains the experiments that display the main features of the method. Finally, the conclusions of the paper are summarized in Section VI.

II. The proposed algorithm: Average of Warped Lines

Our purpose in this Section is to give a brief description of our algorithm, we shall give a more detailed explanation in Section III.

Let $I(i,j,t)$ be an image sequence defined in $\Omega = \{1,\ldots,M\} \times \{1,\ldots,N\} \times \{0,\ldots,T-1\}$. For convenience we shall denote the space image domain by Ω_s, i.e., $\Omega_s = \{1,\ldots,M\} \times \{1,\ldots,N\}$. We shall assume the following image formation model

$$I(i,j,t) = u(i,j,t) + n(i,j,t) \qquad (i,j,t) \in \Omega, \tag{1}$$

[3] As we will see later, this is usually unpractical and similar samples are found in the proximity of the pixel to be processed.

where $u(i,j,t)$ represents the ideal image sequence with no noise, and $n(i,j,t)$ represents a noise perturbation of u. In Section V we will see how our algorithm performs denoising also on sequences where the noise cannot be modeled as additive.

The input to our algorithm is a video sequence of T images (*frames*) I^t, $t = 0, ..., T-1$ where $I^t(i,j) = I(i,j,t)$. We partition the domain of each frame I^t into a disjoint set of lines $\mathcal{L}_i^t$, which for simplicity we take as the horizontal lines $\mathcal{L}_i^t = \{(i,j,t) : 1 \leq j \leq N\}$, so that $\Omega_s \times \{t\} = \cup_{1 \leq i \leq M} \mathcal{L}_i^t$. For each line $\mathcal{L}_i^t$ we consider the family of its neighboring ones, both spatial (nearby lines in the same frame) and temporal (nearby lines from other frames). Let us call this set of neighboring lines $\mathcal{NL}_i^t$. The set of pixels covered by $\mathcal{NL}_i^t$ is the set

$$\cup \mathcal{NL}_i^t = \{(m,l,s) : |i-m| \leq \delta_i, |t-s| \leq \delta_t, 1 \leq l \leq N\} \tag{2}$$

where $\delta_i, \delta_t > 0$ determine the size of this neighborhood. Each line in $\mathcal{NL}_i^t$ is determined by two coordinates m, s in the above neighborhood, and is denoted by $\mathcal{L}_m^s$.

We take each line $\mathcal{L}_m^s \in \mathcal{NL}_i^t$ and *warp* it so as to make it match with $\mathcal{L}_i^t$. A *warping* between the lines $\mathcal{L}_t^i$ and $\mathcal{L}_m^s$ is a map $\varphi_{(m,s)}^{(i,t)} : \mathcal{L}_i^t \to \mathcal{L}_m^s$. Given the warping $\varphi_{(m,s)}^{(i,t)}$ we can transform the image on the line $\mathcal{L}_m^s$, i.e., $I_m^s = I|_{\mathcal{L}_m^s}$ into its warped version on $\mathcal{L}_i^t$ defined by

$$\varphi_{(m,s)}^{(i,t)} \circ I_m^s(j) = I_m^s(\varphi_{(m,s)}^{(i,t)}(j)) = I(m, \varphi_{(m,s)}^{(i,t)}(j), s), \tag{3}$$

if $(i,j,t) \in \mathcal{L}_i^t$. We denote by $\mathcal{W}(I, \mathcal{L}_m^s, \mathcal{L}_i^t)$ the warped version of the image I from the line $\mathcal{L}_m^s$ onto the line $\mathcal{L}_i^t$ which is given by (3).

We compute a warping $\varphi_{(m,s)}^{(i,t)}$ as the displacement such that the images I_i^t and $\mathcal{W}(I, \mathcal{L}_m^s, \mathcal{L}_i^t)$ look as similar as possible. When we have performed this warping operation on every line in the set $\mathcal{NL}_i^t$, we can obtain the denoised version $\mathcal{D}_i^t$ of I on the original line $\mathcal{L}_i^t$ by performing an average (mean or median) of the warped lines $\mathcal{W}(I, \mathcal{L}_m^s, \mathcal{L}_i^t)$ for $\mathcal{L}_m^s \in \mathcal{NL}_i^t$, i.e.,

$$\mathcal{D}_i^t(j) = \text{average}(\mathcal{W}(I, \mathcal{L}_m^s, \mathcal{L}_i^t)(j) : \mathcal{L}_m^s \in \mathcal{NL}_i^t). \tag{4}$$

We do the same for every frame t and line $\mathcal{L}_i^t$ in the video sequence and we obtain the denoised image $\mathcal{D}(i,j,t)$.

We could think in more general terms and apply this warping and averaging procedure to $2D$ regions instead of lines. But we choose the regions $\mathcal{L}_i^t$ to be scan-lines, that is, horizontal lines spanning the whole image from the first to the last column. The reason for this choice is that with $1D$ regions the warping operation reduces to the problem of dense matching in stereo applications, for which there is extensive literature (see for instance [19] and references therein) and, more importantly, we can use dynamic programming to compute a warping between lines which satisfies the *ordering constraint* [4] and a *uniqueness constraint* (when there are no occlusions) [5]. As we have said, this warping can be computed with a dynamic programming algorithm whose complexity is $O(N)$, where N is the number of columns of the image. Let us elaborate on this.

A. Warping through stereo matching

In [19] the authors present a stereo algorithm that, given two corresponding epipolar lines of two views (*left image* and *right image*) of the same scene, finds the set of correspondences that minimize a cost function. The correspondences are pixel pairs (P, P'), where P is a pixel in the *left line* $\mathcal{L}$ and P' is its matching pixel in the *right line* $\mathcal{L}'$. The cost function is a maximum likelihood cost function consisting of a weighted square error term (the squared ℓ^2 distance between the neighborhoods of two pixels) if two pixels are matched or a fixed cost if a pixel cannot be matched. This last case represents that an occlusion happened.

The type of warping computed in [19] satisfies the ordering constraint and is adapted to the presence of occlusions or disocclusions. A *warping* from $\mathcal{L}$ onto $\mathcal{L}'$ (two lines of the image I) that satisfies the ordering constraint is a multi-valued map $\varphi : \mathcal{L} \to \mathcal{S}(\mathcal{L}')$, where $\mathcal{S}(\mathcal{L}')$ denotes the family of subintervals of $\mathcal{L}'$ such that

$$\cup_{j \in \mathcal{L}} \varphi(j) = \mathcal{L}', \tag{5}$$

[4] This constraint states that if pixel P is before pixel Q in line L, then their correspondent matches in line L', pixels P' and Q', will also satisfy that P' comes before Q'.

[5] This constraint states that any pixel P in line L will only match one pixel P' in line L'.

$$\text{if } j < j',\ j, j' \in \mathcal{L}, \text{ then } m \leq m',\ \forall m \in \varphi(j),\ \forall m' \in \varphi(j'). \tag{6}$$

We shall also say in this case that the multi-valued map is *non-decreasing*. We stress the fact that we look for a correspondence map φ which is multi-valued, the reason being the presence of disocclusions. Indeed, if $\varphi(j)$ is an interval (the computational method gives us intervals), it means that the pixel indexed by $j \in \mathcal{L}$ has been transformed in the set of pixels $\{j' \in \mathcal{L}' : j' \in \varphi(j)\}$ in the line $\mathcal{L}'$. With a similar interpretation, it may happen that the interval $J = \{j : j_1 \leq j \leq j_2\}$ is mapped by φ into a point $j' \in \mathcal{L}'$ and this means that the interval J is occluded and mapped to a single pixel $j' \in \mathcal{L}'$.

The inverse of a non-decreasing warping from $\mathcal{L}$ onto $\mathcal{L}'$ is also a non-decreasing warping from $\mathcal{L}'$ onto $\mathcal{L}$, that is our computational method is symmetric with respect to the pair of lines $\mathcal{L}$ and $\mathcal{L}'$. We shall denote by $\mathcal{W}(\mathcal{L}, \mathcal{L}')$ the set of non-decreasing warpings from $\mathcal{L}$ onto $\mathcal{L}'$.

For each (i, j, t) let $\mathcal{N}_{ijt}$ be a two-dimensional neighborhood of (i, j, t) in $\Omega_s \times \{t\}$. We assume that $\mathcal{N}_{ijt} = (i, j, t) + \mathcal{N}$ where $\mathcal{N}$ is a fixed two-dimensional neighborhood of $(0, 0, 0)$. Given I and $\mathcal{N}_{ijt}$, $\mathcal{N}_{mls}$ we define the distance

$$d(\mathcal{N}_{ijt}, \mathcal{N}_{mls})^2 = \sum_{(\alpha,\beta,\gamma)\in\mathcal{N}} |I(i+\alpha, j+\beta, t+\gamma) - I(m+\alpha, l+\beta, s+\gamma)|^2.$$

Let us fix two lines $\mathcal{L}_i^t$ and $\mathcal{L}_m^s$ of the image sequence I. The cost associated to the nondecreasing warping $\varphi : \mathcal{L}_i^t \to \mathcal{S}(\mathcal{L}_m^s)$ is given by

$$\mathcal{C}(\varphi) = \sum_{j\in\mathcal{L}, l\in\varphi(j)} (\kappa_0 + \kappa|j - l|) d(\mathcal{N}_{ijt}, \mathcal{N}_{mls})^2 \tag{7}$$

where $\kappa_0, \kappa \geq 0$. The cost computes the best matching between the lines $\mathcal{L}_i^t$ and $\mathcal{L}_m^s$ where the cost associated to the matching of $j \in \mathcal{L}_i^t$ and $l \in \mathcal{L}_m^s$ is given by the square distance $d(\mathcal{N}_{ijt}, \mathcal{N}_{mls})^2$ weighted by the factor $\kappa_0 + \kappa|j - l|$ which takes into account the horizontal distance from j to l. This weighting function is slightly different to the one used in [19] for which $\kappa = 0$.

We choose the warping between the lines $\mathcal{L}_i^t$ and $\mathcal{L}_m^s$ such that

$$\varphi_{(m,s)}^{(i,t)} \text{ minimizes } \{\mathcal{C}(\varphi) : \varphi \in \mathcal{W}(\mathcal{L}_i^t, \mathcal{L}_m^s)\}\ . \tag{8}$$

In this way, we may define the denoised image $\mathcal{D}(i, j, t)$ by the formula (4), that is, we put in each position j of the denoised line $\mathcal{D}_i^t$ the average of all the matches of j on the lines of $\mathcal{NL}_i^t$.

The optimum in (8) is computed using a dynamic programming algorithm. After computing the matrix of costs whose (j, l) entry is the cost $C(j, l) = (\kappa_0 + \kappa|j - l|)d(\mathcal{N}_{ijt}, \mathcal{N}_{mls})^2$ associated to each correspondence between the pixels $j \in \mathcal{L}_i^t$ and $l \in \mathcal{L}_m^s$, the algorithm proposed in [19] computes the non-decreasing warping that minimizes (7). The warping is represented as a non-decreasing path in matrix $C(j, l)$ and is a list of pixel pairs whose gray values match.

Figure 1 shows, on the top row, two 100 × 100 images which are details of two consecutive frames of the film "The Testament of Dr. Mabuse". On the bottom row of the figure we see the profiles of scanline number 50 in each image. Figure 2 shows, on the left, the central diagonal band of the 100 × 100 cost matrix $C(j, l)$ for matching those lines. In this figure, a light grey value corresponds to a high cost, and a dark grey value corresponds to a low cost. Therefore, matching these lines amounts to finding the path in this matrix (from the top left to the bottom right) which has a minimum accumulated cost. In general, finding the solution to this optimization problem has a complexity of $O(N^3)$, where N is the number of pixels of each line. Introducing the aforementioned constraints of uniqueness and ordering, the complexity reduces to $O(N^2)$. As we see (fig. 2, right) the optimal path does not deviate substantially from the diagonal, allowing us to compute just a diagonal band instead of the whole matrix. Restricting the optimal path to such a band reduces the complexity to $O(N)$. Let us also point out that the total complexity of the denoising procedure is still $O(N)$, since we are performing a number of warping/matching operations which is independent of the size of the image.

III. Analysis and comparison with NL-Means algorithm

As we will see later, our algorithm is adapted to the de-noising of very different types of noise. In order to gain some insight into the reasons for the success of this method, we will compare it with the work of Buades et al.,

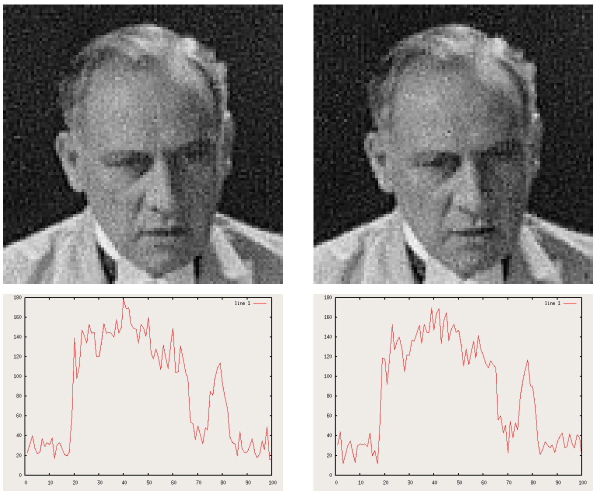

Fig. 1. *Top row:* 100×100 *details of two consecutive frames of the film "The Testament of Dr. Mabuse". Bottom row: corresponding profiles of scanlines number* 50 *in each image.*

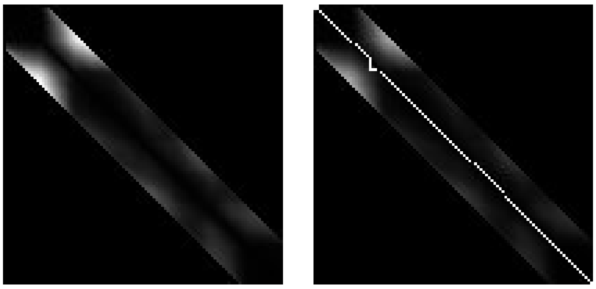

Fig. 2. *Left: cost function for matching line number* 50 *in the images shown in fig. 1. Right: same cost function with optimal path superimposed.*

the Non-local Means (NL-Means) method [14], [15]. For simplicity, let us review their basic method, developed for still images in [14]. In this case, the image model can be written as

$$I(i,j) = u(i,j) + n(i,j) \qquad (i,j) \in \Omega_s\,, \tag{9}$$

where $\Omega_s := \{1,\ldots,M\} \times \{1,\ldots,N\}$, $u(i,j)$ represents the ideal image with no noise, and $n(i,j)$ represents a white noise perturbation of u with zero mean and variance σ^2. The authors of [14] proposed to denoise an image I at the pixel (i,j) by a weighted averaging of the values of the pixels with a neighborhood similar to a given neighborhood of (i,j). The precise formula for NL-Means is

$$\hat{I}(i,j) = \frac{1}{Q(i,j)} \sum_{(m,l)\in\Omega_s} e^{-\frac{(G_a * |I((i,j)+\cdot) - I((m,l)+\cdot)|^2)(0)}{h^2}} I(m,l) \qquad (i,j) \in \Omega_s, \tag{10}$$

where $Q(i,j)$ is a normalization factor given by

$$Q(i,j) := \sum_{(m,l)\in\Omega_s} e^{-\frac{(G_a * |I((i,j)+\cdot) - I((m,l)+\cdot)|^2)(0)}{h^2}}$$

and

$$(G_a * |I((i,j)+\cdot) - I((m,l)+\cdot)|^2)(0) = \sum_{(\alpha,\beta)} G_a(\alpha,\beta)|I(i+\alpha,j+\beta) - I(m+\alpha,l+\beta)|^2.$$

The function G_a denotes a window function which permits to compare the neighborhoods of the pixels (i,j) and $(m.l)$. In practice one can take the discretization of a Gaussian function of standard deviation a, or the characteristic function of a neighborhood of $(0,0)$. In what follows, we denote the neighborhood of each pixel (m,l) by $\mathcal{N}_{ml}$ so that $\mathcal{N}_{ml} = (m,l) + \mathcal{N}_{00}$. Notice that rapid decreasing of the exponential formula in (10) makes that only pixels (m,l) with a neighborhood similar to the neighborhood of (i,j) contribute to the sum in (10).

In some sense, they look for realizations of the same random variable and assuming that the noise is uncorrelated at different positions, the averaging process reduces the noise variance. The image model (9) contains the contributions of a deterministic process, the function u, and the noise $n(i,j)$ which we are assuming to be a white noise of zero mean and variance σ^2. The comparison of the two random vectors given by the values of $\{I(\alpha,\beta)\}_{(\alpha,\beta)\in\mathcal{N}_{ij}}$ and $\{I(\alpha,\beta)\}_{(\alpha,\beta)\in\mathcal{N}_{ml}}$ serves to identify pixels where the deterministic part of the image is similar, and the variations are mainly due to the random nature of the noise. To understand the noise reduction effect of formula (10) we may consider the noise reduction effect produced by the convolution of I with a Gaussian window of standard deviation $\frac{h}{\sqrt{2}}$. If noise is uncorrelated, averaging reduces its variance according to the formula

$$\mathrm{var}(G_{\frac{h}{\sqrt{2}}} * n(0,0)) = \mathrm{var}\left(\sum_{(\alpha,\beta)\in\mathcal{N}_{00}} G_{\frac{h}{\sqrt{2}}}(\alpha,\beta)n(\alpha,\beta)\right) = \sigma^2 \sum_{(\alpha,\beta)\in\mathcal{N}_{00}} G_{\frac{h}{\sqrt{2}}}(\alpha,\beta)^2 \approx \frac{\sigma^2}{2\pi h^2}.$$

Thus, in this case the noise reduction is equivalent to the one obtained by averaging on a circular neighborhood of radius h which contains approximately $2\pi h^2$ pixels. As a basic rule of thumb this theoretical estimate says that averaging the values of k pixels with similar neighborhoods we can reduce the variance of the noise by a factor $\frac{1}{k}$. In practice, this reduction factor is not attained.

The above formula (10), of complexity $O(N^2)$ (where N stands for the total number of pixels in the image) can be extended to image sequences by extending the sum on the indices (i,j,t). In that case, for each pixel (i,j,t) we have to explore the whole image sequence and the formula becomes extremely cumbersome. This has been circumvented in several ways in [14] and [20], in particular, by restricting the window where pixels will be compared, by a multiscale strategy or dividing the image in blocks [14].

In our proposed method of Average of Warped Lines (AWL) we develop a similar strategy: we consider the pixel (i,j,t) as belonging to a row of the image and we enlarge the set of pixels to be compared to (i,j,t) to those which belong to neighboring lines in space and time, i.e., we consider as search space the set of pixels

$$\cup\mathcal{NL}_i^t := \{(m,l,s) : |i-m| \leq \delta_i,\ |t-s| \leq \delta_t,\ 1 \leq l \leq N\} \tag{11}$$

for some $\delta_i, \delta_t > 0$. Then we take the average of the values of the pixels in $\cup\mathcal{NL}_i^t$ whose neighborhood is similar to the neighborhood of (i,j,t). The crucial difference is that we include in this process a further geometric constraint determined by the level set structure of the image. Let us give a more detailed description of this strategy.

To explain the basic idea, let us concentrate on 2D images. Recall that the upper (resp. lower) level sets of an image $v : \Omega_s \to \{1,\ldots,L\}$ are the sets

$$X^\lambda v = \{(m,l) : v(m,l) \geq \lambda\} \qquad (\text{resp. } X_\lambda v = \{(m,l) : v(m,l) < \lambda\}),$$

where $\lambda \in \{1,\ldots,L\}$. We call level lines the boundaries of the connected components of level sets. Using the right notion of connectivity (8-connectedness for upper level sets and 4-connectedness for lower level sets), the level lines do not cross each other. We propose to interpret this constraint in our context.

Suppose that we want to filter the image I at the pixel (i,j) following the basic ideas described above. If $\mathcal{N}_{ij}$ is a neighborhood of (i,j), we want to find the pixels $\{(m(i,j),l(i,j))\}$ whose neighborhood is similar to $\mathcal{N}_{ij}$, and then we shall average their values (with a simple average formula or using a formula similar to (10)). On one hand, for computational reasons we restrict our search to pixels (m,l) which are in a neighborhood of (i,j). On the other hand, we want to impose the geometric constraint given by the inclusion property of the level lines. We can do this if we consider as search space the pixels

$$\cup\mathcal{NL}_i := \{(m,l) : |i-m| \leq \delta,\ 1 \leq l \leq N\},$$

for some positive integer δ, that is, the rows which are near to the row i containing (i,j). In that case, if (i,j') is another pixel in the row i and $(m(i,j),l(i,j))$, $(m(i,j'),l(i,j'))$ are two pixels that are similar to (i,j) and (i,j') respectively, with $m(i,j) = m(i,j')$, the geometric constraint imposed by the inclusion of level lines is translated into the fact that "$j < j'$ implies that $l(i,j) \leq l(i,j')$". To do this we compute correspondences of all points on a row at the same time and search for a non-decreasing *warping* $\varphi_{i,i'}$ of row i onto row i' (i.e., satisfying (5) and

(6)) which gives for each pixel (i,j) its similar or similar ones in the row i'. We stress again the fact that the correspondence map $\varphi_{i,i'}$ is multivalued, the reason being the presence of disocclusions.

The set of points $\mathcal{W}_{ij} := \{(m,l) : |i-m| \leq \delta,\, l \in \varphi_{i,m}(j)\}$ is the set of points whose neighborhoods are similar to $\mathcal{N}_{ij}$. Our algorithm will replace the value $I(i,j)$ by an average of the values of the pixels in $\mathcal{W}_{ij}$. A general averaging formula can be developed following the arguments of [21]. For each $(m,l) \in \mathcal{W}_{ij}$, let us denote the matching cost of the neighborhoods of both pixels by

$$C((i,j),(m,l)) = (\kappa_0 + \kappa|j-l|)d(\mathcal{N}_{ij},\mathcal{N}_{ml})^2.$$

Then we define the reliability as a decreasing function of the cost, in particular, we may use

$$\epsilon((i,j),(m,l)) = e^{-C((i,j),(m,l))}.$$

Then the estimated value of the denoised image $u(i,j)$ is the value of $\mathcal{I}$ that minimizes the quantity

$$\sum_{(m,l)\in\mathcal{W}_{ij}} \epsilon((i,j),(m,l))|\mathcal{I} - I(m,l)|^p.$$

When $p = 2$, the optimal value is

$$\hat{I}(i,j) = \frac{\sum\limits_{(m,l)\in\mathcal{W}_{ij}} \epsilon((i,j),(m,l))I(m,l)}{\sum\limits_{(m,l)\in\mathcal{W}_{ij}} \epsilon((i,j),(m,l))}. \tag{12}$$

This algorithm when $p = 2$ was proposed in [21] to obtain the estimated value of a pixel to be inpainted using the pixels with similar neighborhoods. Notice that this formula coincides with (10) if the cost function is minus the argument of the exponential in (10). When $p = 1$, then $\hat{I}(i,j)$ is the median of the values $I(m,l)$, $(m,l) \in \mathcal{W}_{ij}$, with respect to the probability distribution given by the normalized reliability values $\frac{\epsilon((i,j),(m,l))}{\sum\limits_{(m,l)\in\mathcal{W}_{ij}} \epsilon((i,j),(m,l))}$.

For computational simplicity it looks better to use an averaging formula with uniform weights instead of (12). The corresponding theoretical reduction of the variance of the noise will depend on the number of points in $\mathcal{W}_{ij}$ according to the formula

$$\operatorname{var}\left(\frac{\sum_{(m,l)\in\mathcal{W}_{ij}} I(m,l)}{\operatorname{card}(\mathcal{W}_{ij})}\right) = \frac{\sigma^2}{\operatorname{card}(\mathcal{W}_{ij})}. \tag{13}$$

As an heuristic interpretation of AWL, we can say that we estimate the level line going trough the pixel (i,j) and we average the actual values of the corresponding pixels. If the estimate is correct the values $I(i,j)$ and $I(m,l)$ with $(m,l) \in \mathcal{W}_{ij}$ differ only in its noise component and its average will keep the image structure while it will reduce the noise according to (13). This interpretation suggests the following observation: if the level line is not transversal to the rows of the image, i.e., it is tangent to the horizontal direction, then it would be better to consider the rows in the vertical direction to cut the level line transversally and get enough correspondent points for a given pixel. This may be necessary in some cases. In particular, this situation happens for flat regions (there are no level lines) and using two orthogonal directions leads to a more isotropic averaging in those pixels. This is corroborated by our experiments below.

The above proposal can be naturally extended to the case of movies or video sequences assuming that the image model is given by (1). We may define the upper (resp. lower) level sets of the image sequence $v(i,j,t)$ as the sets

$$X^\lambda v = \{(m,l,s) : v(m,l,s) \geq \lambda\} \qquad (\text{resp. } X_\lambda v = \{(m,l,s) : v(m,l,s) < \lambda\}).$$

We call level surfaces the boundaries of connected components of level sets. Using the right notion of connectivity, the level surfaces are embedded into each other. For each voxel (i,j,t) and its neighborhood $\mathcal{N}_{ijt}$, we look for pixels (m,l,s) in the set $\cup\mathcal{NL}_i^t$ given in (11) with a neighborhood similar to $\mathcal{N}_{ijt}$. As in the case of 2D images, we map the row $(i,\cdot,t)$ to a row $(m,\cdot,s)$ through the multivalued non-decreasing warping $\varphi_{(m,s)}^{(i,t)}$ and we define $\mathcal{W}_{ijt} := \{(m,l,s) : |i-m| \leq \delta_i\, , |t-s| \leq \delta_t,\, l \in \varphi_{(m,s)}^{(i,t)}(j)\}$ as the set of pixels with a neighborhood similar to $\mathcal{N}_{ijt}$. The nondecreasing nature of $\varphi_{(m,s)}^{(i,t)}$ translates the geometric constraint on the inclusion structure of the level

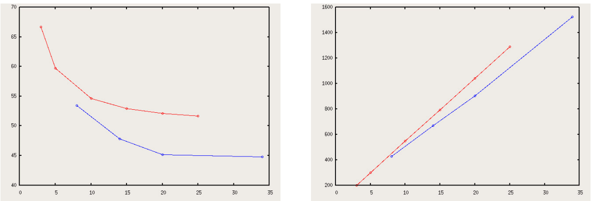

Fig. 3. *Left image: average, over 10 consecutive frames of the sequence in fig. 12, of the MSE between the original and the de-noised sequence. Upper curve: MSE for NL-Means. Lower curve: MSE for AWL. The X axis denotes the number k of pixels (NL-Means) or lines (AWL) chosen for the average. See text. Right image: $MSE \times k$ vs.. k, for NL-Means (upper curve) and AWL (lower curve). Notice that the linear dependence shown implies that the MSE decays with the number k of* averaging neighbors *as $\frac{1}{k}$ for both methods.*

surfaces of the image. Finally, the average of I on the pixels in $\mathcal{W}_{ijt}$ will give our estimate of the denoised image $u(i,j,t)$.

Again, as an heuristic interpretation of this algorithm, we can say that we estimate the level surface going trough the pixel (i,j,t) and we average the actual values of the corresponding pixels. If the estimate is correct the values $I(i,j,t)$ and $I(m,l,s)$ with $(m,l,s) \in \mathcal{W}_{ijt}$ differ only in its noise component and its average will keep the image structure while it will reduce the noise.

In order to test the Equation (13) we processed the sequence in Figure 12 with a different number of points. In the case on NL-Means, for each pixel to be denoised we considered its k spatio-temporal nearest neighbors. For AWL we selected k as the number of lines used. The results are depicted in Figure 3 where we show the evolution of the MSE with k. Both, for NL-Means and AWL, the MSE decreases as k increases, even if the theoretical estimate (13) does not hold (the MSE is not divided by 2 when we double the number of points k).

IV. Numerical implementation

For the warping we have adapted the *Maximum Likelihood Minimum Horizontal Discontinuities* version of the stereo algorithm of Cox et al. presented in [19], making two changes on the computation of the cost function. Firstly, to compute the element $C(j,l)$ of the cost matrix C, we compare not the isolated pixels j and l but rather their spatial $k \times k$ neighbourhoods[6]: we compute the Sum of Squared Differences (SSD) of these neighborhoods. Secondly, we multiply this SSD by a factor of the form $(0.9 + 0.1|j-l|)$ in order to penalize matches of far away (high $|j-l|$) pixels (see the expression after the sum in formula (7)). We have found that these changes noticeably increase the robustness of the matching procedure. To speed up the process we may look for the optimal path in a band of constant width around the diagonal of the matrix cost. In our experiments, the width of the diagonal band on which we compute the cost function has been taken as the 10 per cent of the size of the matrix.

For each line $(i,\cdot,t)$ that we want to denoise we consider a spatio-temporal neighborhood $\cup \mathcal{NL}_i^t$ with a total of $r \times n$ neighboring lines: r neighboring lines per frame over n neighboring frames. These two values r and n are the two only parameters of our algorithm. Also, we may choose which sort of average we want to perform on the matching values, mean or median average.

Note that our parameters do not depend on the nature of the noise present in the image. This is a very important feature of our method, the fact that we do not need to model the noise in order to be able to remove it.

V. Experiments

We have implemented our algorithm in $C++$ under Linux, on a $1GHz$ $1GbRAM$ PC, with non-optimized code. For a 240×136 color video sequence and averaging over neighborhoods of $r=3$ lines per frame and $n=5$ frames, the computational time is roughly 2 seconds per frame.

To show the effectiveness of the warping included in the proposed method we present the results for the image in Figure 4. This is the middle frame of a 19 frame long sequence depicting a circle moving on a uniform background, with added Gaussian noise of variance $\sigma = 3$. In Figures 4 and 5 we display the results both of AWL and NL-Means and we compare the MSE error (between the result and the original image without noise). In the first experiment (left and middle left images), for the NL-Means method we used the parameters suggested in [14] over 5 frames.

[6] In practice we took $k=11$, but results do not vary noticeably for k=9,11,13

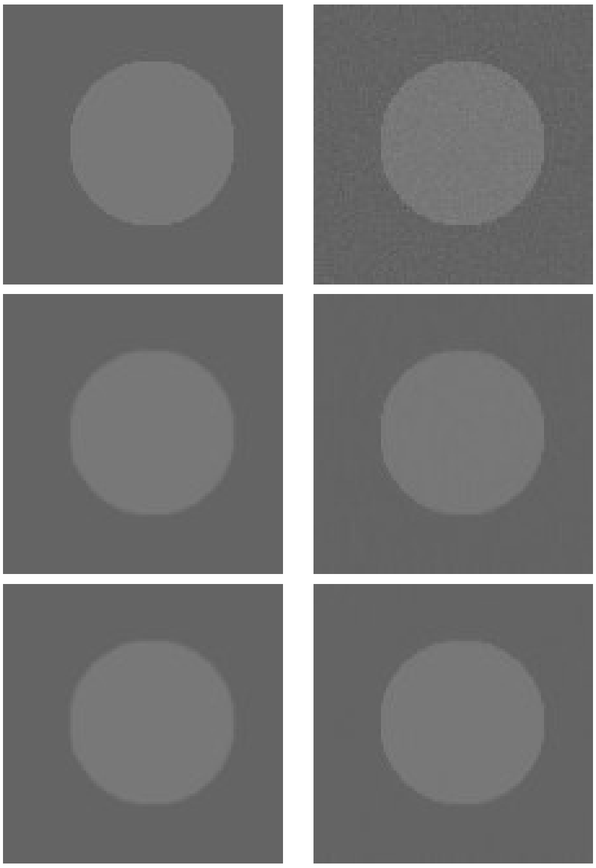

Fig. 4. *Top row. Left: original frame. Right: original plus Gaussian noise of $\sigma = 3$. Middle row. Left: denoised with NL-Means using 5 frames. Right: denoised with AWL using a search space of 5 frames and 5 lines per frame. Bottom row. Left: denoised with NL-Means using 19 frames. Right: denoised with AWL using a search space of 19 frames and 5 lines per frame.*

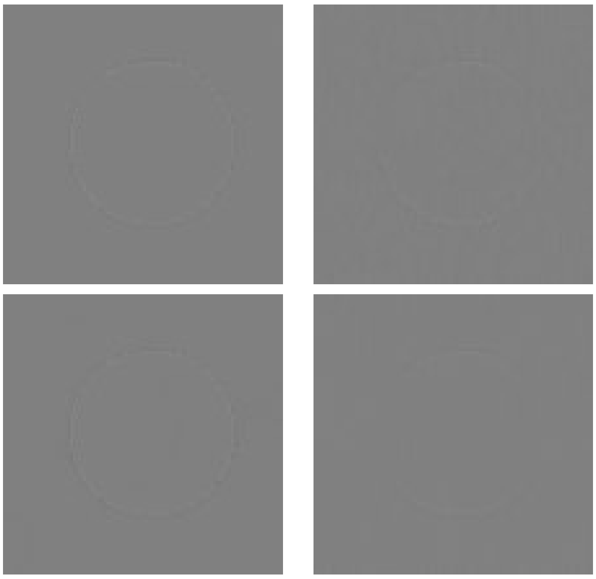

Fig. 5. *Top row. Left: difference between original in fig. 4 and denoised with NL-Means using 5 frames ($MSE = 0.73$). Right: difference between original and denoised with AWL using 5 frames($MSE = 0.92$). Bottom row. Left: difference between original and denoised with NL-Means using 19 frames ($MSE = 0.73$). Right: difference between original and denoised with AWL using 19 frames ($MSE = 0.37$).*

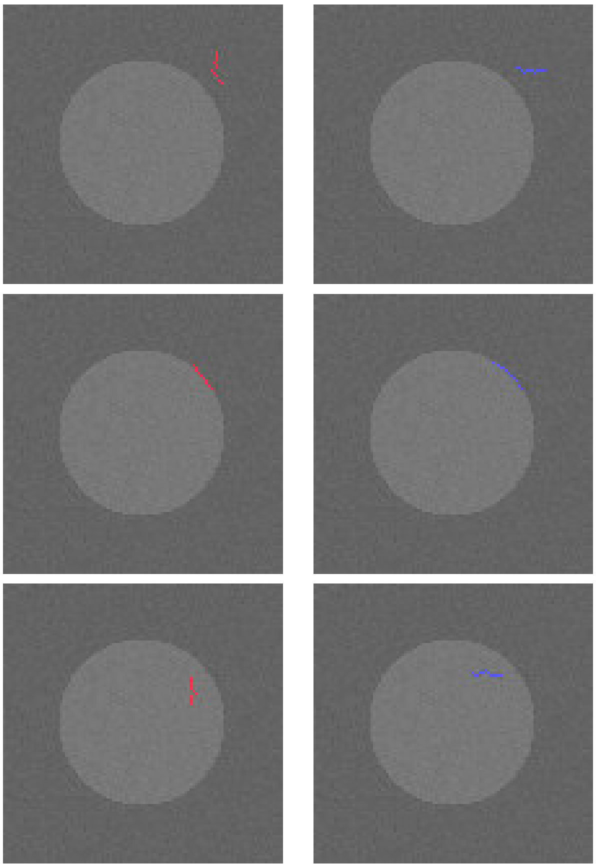

Fig. 6. *Several matching sets shown as superimposed curves. Each curve is the set of matches found for its middle point. Left: AWL with horizontal lines. Right: AWL with vertical lines.*

The parameters for AWL were: search space of 5 frames and 5 lines per frame, median average. In the second experiment (middle right and right images), for the NL-Means method we used the parameters suggested in [14] over 19 frames. The parameters for AWL were: search space of 19 frames and 5 lines per frame, median average. First, we observe that in both experiments, AWL did not blur the edges while NL-Means did. This is better displayed in Figure 5 where we see that the edges of the circle are better preserved with AWL. A constant is added to the results for better visualization. Second, we can see that the MSE for NL-Means did not improve when increasing the number of frames considered, while AWL monotonically decreases the MSE to achieve a MSE of 0.37.

Finally, in Figure 6 we depict the set of matches found for some points. Each red or blue curve represents the set of matches found for its middle point. We see that the matches tend to follow the level line when there is one. In flat regions it would be better to use all points obtained looking for matches in horizontal and vertical lines, thus simulating a Laplacian diffusion.

Figure 9 shows on the top left a frame from a film which is heavily corrupted with noise. On the top right we see the result of applying Field of Experts (FoE) denoising [1] to this image. On the bottom left we see the result of the NL-Means denoising algorithm. On the bottom right we see the result obtained with our Average of Warped Lines (AWL) technique. The FoE denoising result is a bit over-smoothed: notice the crystal jar in the background, the square tiles, the writing on the sheet of paper on the desk. Also, this method seems to enhance the speckle (impulse) noise: notice this effect on the face of the actor, the desk, near the legs, etc. Concerning the NL-Means result, it looks clean and sharp, but some parts have been oversmoothed, like the above mentioned square tiles behind the desk or the writing on the paper sheet on the desk. These structures are better respected in the AWL result. On the other hand, there is another distracting visual artifact present in the NL-Means result which is only noticeable by watching the denoised video. While one frame at a time the images denoised with NL-Means look perfectly reasonable, when we see them *in motion* at a rate of $24fps$ we clearly perceive an oscillation in the boundary of the objects in the scene. In some cases this artifact really catches one's eye, like when the lamp in this scene, which we know to be rigid and static, slightly deforms and seems to be bending from one frame to the next. Figure 8 points this out by showing a detail (corresponding to the lamp just mentioned) of the difference between the original and denoised images both for NL-Means and AWL. Observe that the boundary of the lamp is clearly visible for the NL-Means case. This means that the process of NL-Means denoising has introduced an error of order $0.5 - 1.0$ pixels in the structures present in the image, and while this error is acceptable for still images it is not for moving images, where it immediately calls the viewer's attention.

The cases of oversmoothing of the NL-Means algorithm may be explained by the fact that it uses all pixels of the image (or a block around the given pixel) in the averaging process. Although each pixel is weighted by its similarity to the pixel to be denoised, if the number of outlier pixels is greater than the number of correct similar pixels the result will be biased. This is the case along edges where a difference of one pixel normal to the edge may produce a small distance and therefore a big weight in (10). In Figure 7 we show the circle image together with the evolution of the MSE depending on the maximum distance for the pixels used in formula (10). Given the set of pixels in the searching area we sort them according to its similarity to the reference pixel. Then we consider a set of similarity thresholds and produce a denoised version of the image considering in formula (10) only those pixels with similarity below the given threshold. Then we can look for the similarity threshold that gives the smaller MSE. One one hand, this shows us that there exist an optimum and on the other it justifies why NL-Means produces oversmoothed results in some cases. Obviously, this experiment can be done only if we have the original image without noise in order to compute the true MSE.

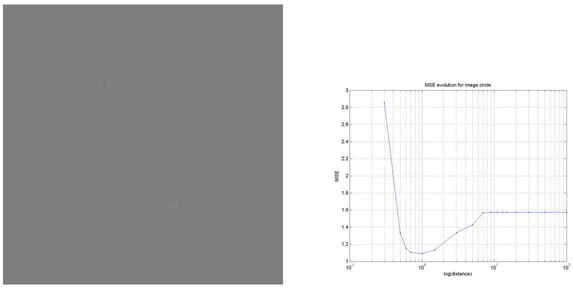

Fig. 7. *Optimal denoised image by NL-Means (left) (MSE = 1.09), MSE evolution depending on the distance of the pixels used.*

Figure 12 shows, on the top row, several frames of the *Flower garden* sequence. The second row shows the result of adding Gaussian noise of $\sigma = 10$ to this sequence. The third row shows the NL-Means de-noising results. The fourth row shows the AWL de-noising results. The Mean Square Error (MSE) between the *original* and each de-noised sequence is comparable: 47.8 for AWL, 49.3 for NL-Means. If we look at some fine texture details in these images, like the tree branches in the background, the bushes and the soil in the foreground, we observe that these structures are better preserved by the AWL algorithm while they are blurred by the NL-means algorithm. We have run NL-Means with the choice of parameters suggested by the authors in their paper. For AWL the parameters were: search space of $n = 5$ frames and $r = 3$ lines, median average.

Figure 10 shows an application of the AWL algorithm to still images. Notice that this is a harder task for AWL

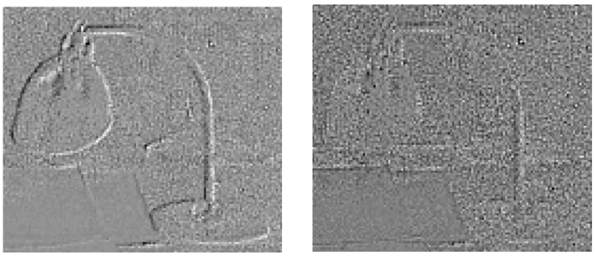

Fig. 8. *Left: difference between original in fig. 9 and denoised with NL-Means (detail, lamp.) Right: difference between original and denoised with AWL (detail, lamp.)*

since the number of similar pixels to a given one which are on a nearby estimated level line is smaller, and we propose a slight modification of the procedure. For the de-noising to be significant we need to average over a number of lines depending on the noise power, typically 15 for the images in our experiments. But if those 15 lines must come from the same image, which is the case of course in still images, then running the AWL may produce some artifacts. These artifacts are noticeable on flat regions of Figure 10, where the matching (for the *warping* procedure) is quite arbitrary, and are seen as vertical streaks, see middle left image. If we apply AWL but on the columns rather than on the scan-lines (rows) of the image, then we get horizontal streaks, see middle right image. Averaging both results, we get a good de-noising with AWL, see right image. So we conclude that with this modification (namely, average of horizontal-AWL and vertical-AWL) we can perform de-noising on still images with AWL. This effect has been explained in Section III: for the AWL to be effective we need that the level line cuts transversally the system of lines used in the warping process. This cannot be the case in flat regions where there are no level lines and using two orthogonal directions leads to a more isotropic averaging in those pixels. If the noise is quite significant, though, just averaging horizontal and vertical AWL might not be enough, and we should consider other directions as well, e.g. average of AWL with lines with a slope of $0, 45, 90$ and 135 degrees. Or adapting our proposed approach to the warping of $2D$ regions instead of lines. This is the subject of our future research.

Figure 11 shows that the performance of AWL does not necessarily decay when there is a change of scene in the movie. One could think that, in the case that frame t of the movie marks a transition from one scene to another, then pixels in frames $t-1$, t, $t+1$ would have far less matches in neighboring frames and the de-noising results would therefore be hindered. But this does not happen in the experiment of Figure 11, as we can see in the bottom row of this figure. The search space is of $n = 5$ frames and $r = 3$ lines. In any case, the application of the method requires a detection of scene change to avoid possible artifacts in this case.

Figure 13 shows, on the top row, several frames of a 128×96 video captured with a mobile phone camera. The compression artifacts are clearly noticeable, and a completely different type of noise than that present in Figure 9. In the middle row we see the results obtained with AWL using mean average. In the bottom row we see the results obtained with AWL using median average. In both cases we have chosen neighborhoods of $r = 3$ lines per frame and $n = 3$ frames. The denoising results look good and quite similar, maybe in the case of the median average they are a bit sharper (look for instance at the area around the eyes.)

Figure 14 shows, on the top row, several frames of a film compressed with MPEG4 at a bitrate of 128kbps. In the bottom row we see the results obtained with AWL using mean average ($r = 3$ lines per frame and $n = 5$ frames). We can see that the blocking artifacts have mostly disappeared. Since in this case we have the uncompressed (noiseless) original material, we can compare in Figure 15 our result with the original frame compressed at double the bitrate, $256kbps$. The $128kbps$ frame denoised with AWL has less noticeable blocking artifacts than the $256kbps$ frame, while on the other hand it is less sharp.

VI. Conclusions and future work

We have introduced in this paper a new algorithm for denoising of image sequences. The algorithm does not require any motion estimation and is based on the well-known fact that averaging several realizations of a random variable reduces noise variance. Assuming that the observed image sequence I is the sum of an ideal denoised image plus some white noise (see (1)), given the pixel position (i, j, t) we estimate the nearby points on the level surface passing trough it, we consider them as realizations of the same random variable and we take an average of them. For that we have adapted an efficient algorithm for computing correspondence maps of epipolar lines in pairs of stereo images which satisfies the ordering constraint (see [19]) which in our case is interpreted as the inclusion ordering of level surfaces. Given a line $\mathcal{L}$ on the image, and considering nearby lines the algorithm computes a warping of $\mathcal{L}$ onto each of them by minimizing a similarity cost between neighborhoods of given pixels along the lines and keeping the ordering constraint. The correspondences associated to each pixel (i, j, t) are interpreted as the estimated points on the level surface through it and the values of I on them as realizations of the same random variable. We obtain the denoised sequence by averaging them. We call the method Average of Warped Lines (AWL). The algorithm is implemented using a dynamic programming principle which is very efficient to compute the desired correspondences. The original complexity of the algorithm per frame is $O(N^2)$, where N is the number of columns of the image, but with a further reasonable simplification it may be reduced to $O(N)$. In practice, for a 240×136 color image sequence, the computational time is 2 seconds per frame on a 1Gh 1GbRAM PC. This feature compares favorably to other recent algorithms like [15], [1]. Finally, we have displayed several experiments to show the performances of the algorithm and to compare it with the algorithms in [15], [1].

The proposed principle and algorithm can be also applied to static images, though this requires some further discussion since we may not have enough redundancy in a neighborhood of a given pixel in a static image to get an effective reduction of noise power. We have displayed a first experiment in this case but we believe that this deserves further discussion to be conducted in a future work.

ACKNOWLEDGMENTS

Many thanks to Guillermo Sapiro for his suggestions and the chance to have very fruitful discussions. Thanks to Andrés Pardo for providing us with the Super-8 sequence. This work originated at and was inspired by the IMA Workshop: The Mathematics and Art of Film Editing and Restoration, Minnesota, February 6 - February 10, 2006. Many thanks to T. Buades and M. Black for providing us with their de-noising results. The first two authors acknowledge partial support by PNPGC project, reference BFM2003-02125, and by IP-RACINE Project. M. Bertalmío acknowledges support by the Ramón y Cajal Program. V. Caselles acknowledges partial support by the Departament d'Universitats, Recerca i Societat de la Informació de la Generalitat de Catalunya. A. Pardo acknowledges the support by Proyecto PDT/S/C/OP/17/07. The black and white film examples are from Fritz Lang's 1933 feature "The Testament of Dr. Mabuse," courtesy Criterion Collection, New York (DVD 2004). Frames from the film "Ruido" (figs. 14 and 15) are courtesy of Lavorágine Films.

REFERENCES

[1] T. M. Moldovan, S. Roth, and M. J. Black, "Denoised archival films using a learned bayesian model," in *ICIP2006*, 2006.

[2] M. Nikolova, "Local strong homogeneity of a regularized estimator," *SIAM J. Appl. Math.*, vol. 61, pp. 633–658, 2000.

[3] A. Buades, B. Coll, and J. M. Morel, "The staircasing effect in neighborhood filters and its solution," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1499–1505, June 2006.

[4] A. Tekalp, *Digital Video Processing*. Prentice Hall, 1995.

[5] B. Alp, P. Haavisto, T. Jarske, K. Oistamo, and Y. Neuvo, "Median based algorithms for image sequence processing," in *Proceedings SPIE Visual Comm. and Image Proc. '90*, 1990, pp. 122–134.

[6] G. Arce, "Multistage order statistic filters for image sequence processing," *IEEE Transactions Signal Processing*, vol. 39, pp. 1146–1163, 1991.

[7] E. Dubois and S. Sabri, "Noise reduction in image sequences using motion compensated temporal filtering," *IEEE Transactions on Communications*, vol. 32, no. 7, pp. 826–832, July 1984.

[8] R. McMann, S. Kreinik, J. Moore, , A. Kaiser, and J. Rossi, "A digital noise reducer for encoded ntsc signals," *SMPTE Journal*, vol. 87, pp. 129–133, March 1979.

[9] R. Samy, "An adaptive image sequence filtering scheme based on motion detection," in *SPIE*, 1985, pp. 135–144.

[10] M. Sezan, M. Ozkan, and S. Fogel, "Temporally adaptive filtering of noisy image sequences using a robust motion estimation algorithm," in *Proc. IEEE Int. Conf. Acoust. Speech, Signal Processing*, 1991, pp. 2429–2432.

[11] J. Lee, "Digital image smoothing and the sigma filter," *Computer Vision, Graphics and Image Processing*, vol. 24, pp. 255–269, 1983.

[12] D. Kuan, A. Sawchuk, T. Strand, and P. Chavel, "Adaptive noise smoothing filter for images with signal-dependent noise," *IEEE Transcations Pattern Anal. Mach. Intell.*, vol. 7, pp. 165–177, March 1985.

[13] M.K.Ozkan, M. Sezan, and A. Tekalp, "Adaptive motion compensated filtering for noisy image sequences," *IEEE Transcations Circuits and Systems for Viedeo Technology*, vol. 3, pp. 277–290, August 1993.

[14] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *SIAM Multiscale Modeling and Simulation*, vol. 4, no. 2, pp. 490–530, 2005.

[15] ——, "Denoising image sequences does not require motion estimation," in *Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance*, 2005, pp. 70–74.

[16] J. Boulanger and C. K. andP. Bouthemy, "Adaptive space-time patch-based method for image sequence restoration," in *Workshop on Statistical Methods in Multi-Image and Video Processing (SMVP'06)*, May 2006.

[17] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *IEEE Int. Conf. on Computer Vision, ICCV'99*, 1999, pp. 1033–1038.

[18] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, June 2005, pp. 860–867.

[19] I. J. Cox, S. L. Hingorani, and S. B. Rao, "A maximum likelihood stereo algorithm," *Computer Vision and Image Understanding*, vol. 63, no. 3, pp. 542–567, May 1996.

[20] M. Mahmoudi and G. Sapiro, "Fast image and video denoising via nonlocal means of similar neighborhodds," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 839–842, December 2005.

[21] Y. Wexler, E. Schechtman, and M. Irani, "Space-time video completion," in *Proceedinggs of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, 2004.

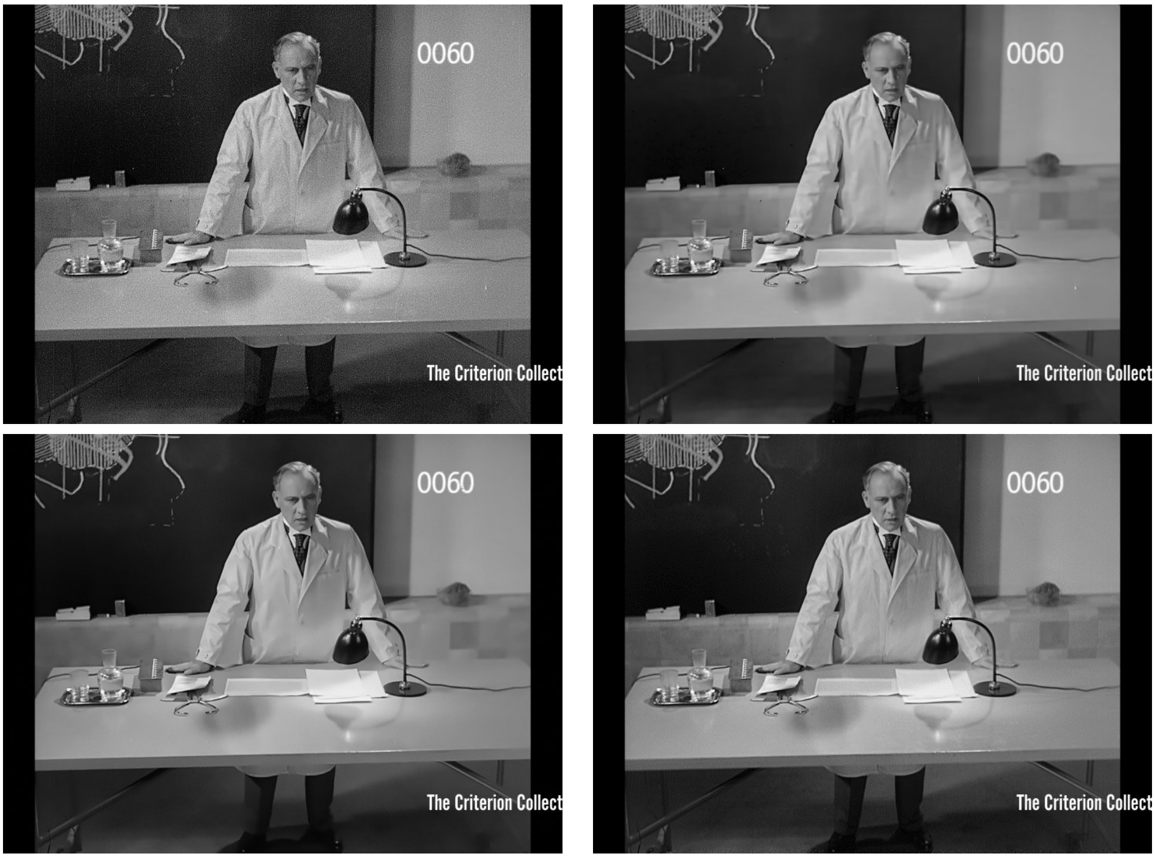

Fig. 9. *Original image (top left), denoised by FoE denoising (top right), denoised by NL-Means (bottom left), our result (bottom right) .*

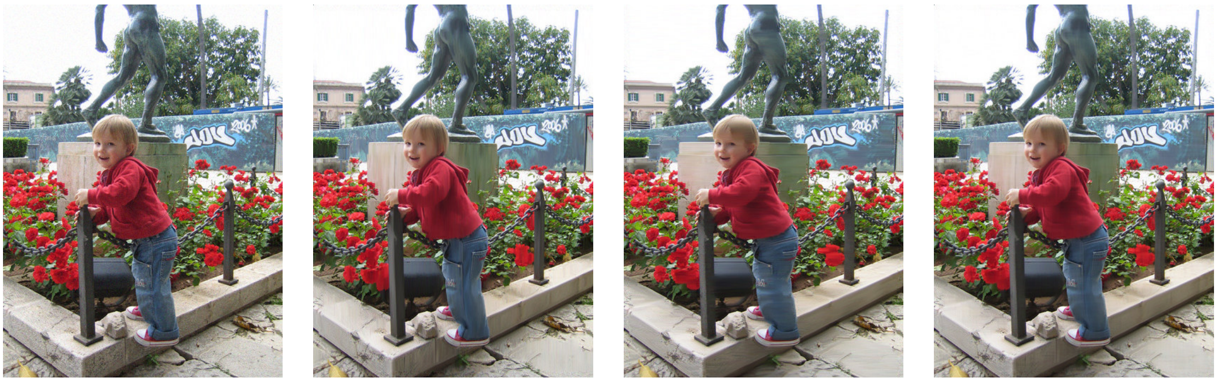

Fig. 10. *De-noising of a still image. Original image (left,) result of AWL applied to horizontal lines (middle left,) result of AWL applied to vertical lines (middle right,) average of vertical and horizontal AWL (right.)*

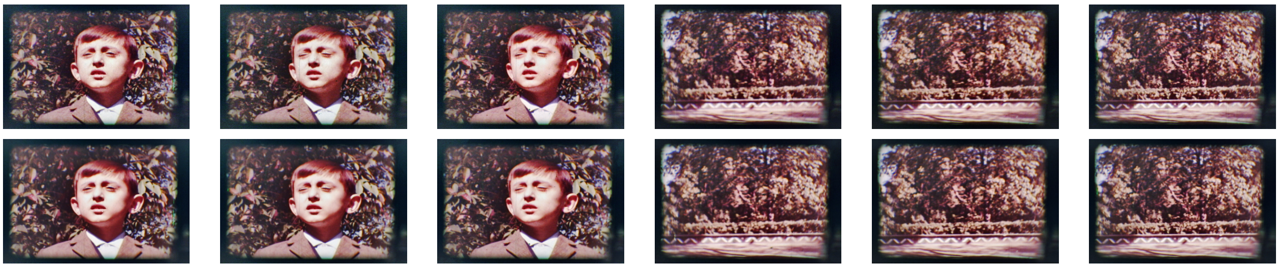

Fig. 11. *Original Super-8 film with shot transition (top,) de-noised with AWL (bottom.) Notice how performance does not decay on the transition.*

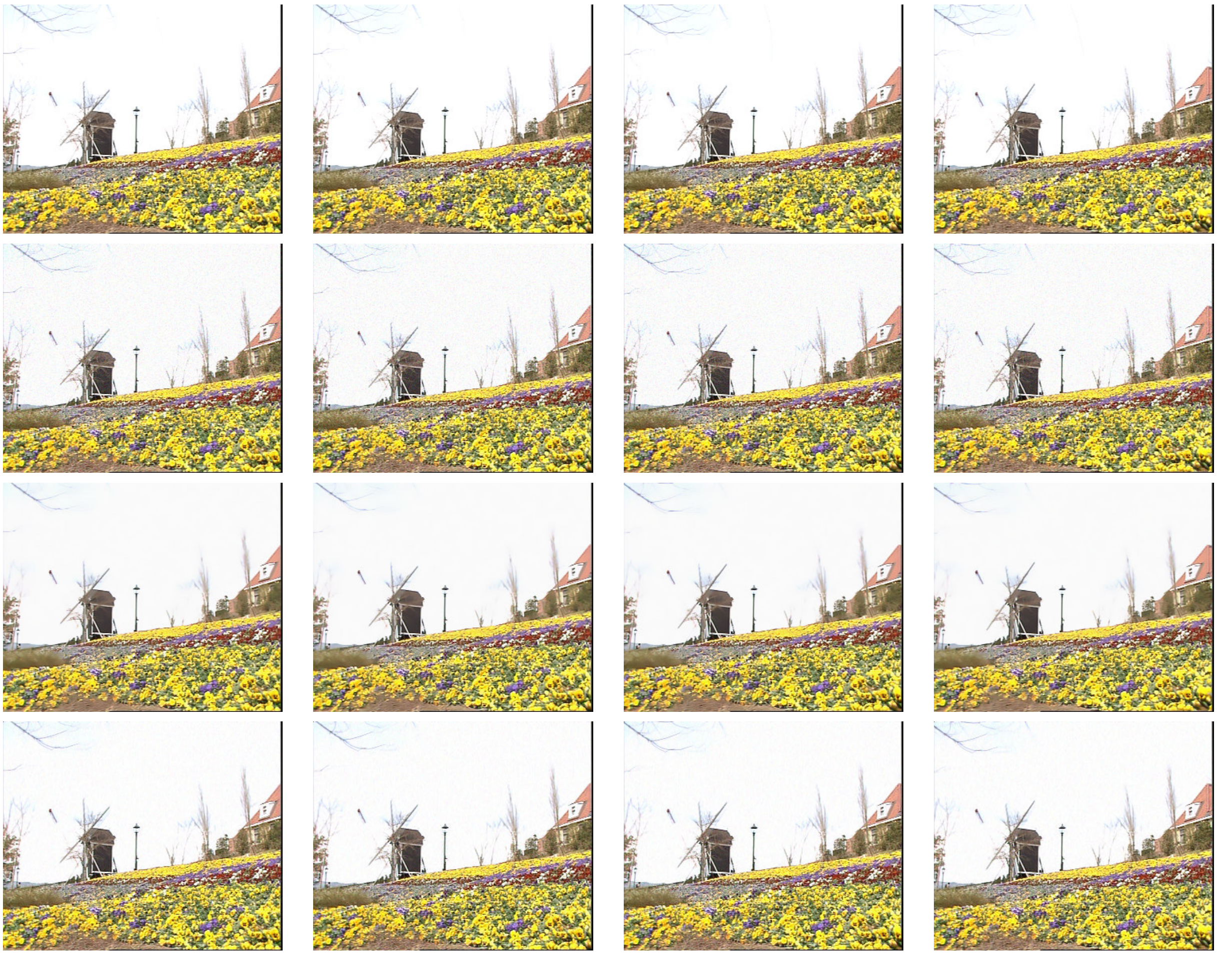

Fig. 12. *Original sequence (top row,) added noise of $\sigma = 10$ (middle top,) denoised with NL-Means (middle bottom,) denoised with AWL using median average (bottom row.)*

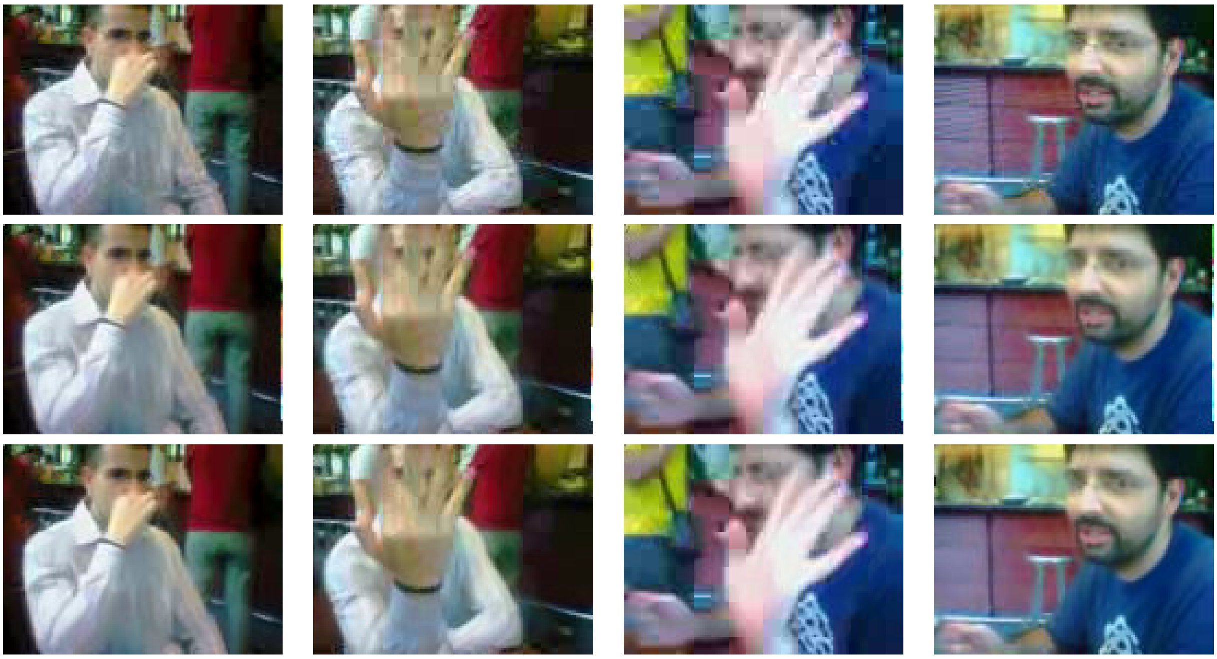

Fig. 13. *Several frames of a video captured with a mobile phone camera (top row), denoised with AWL using mean average (middle row,) denoised with AWL using median average (bottom row.)*

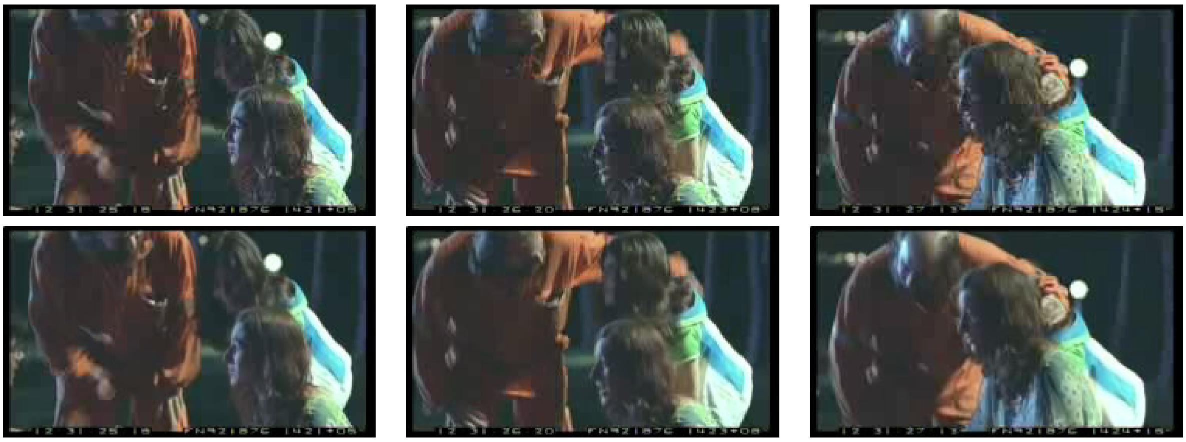

Fig. 14. *Several frames of a film compressed with MPEG4 at 128kbps (top row), denoised with AWL (bottom row.)*

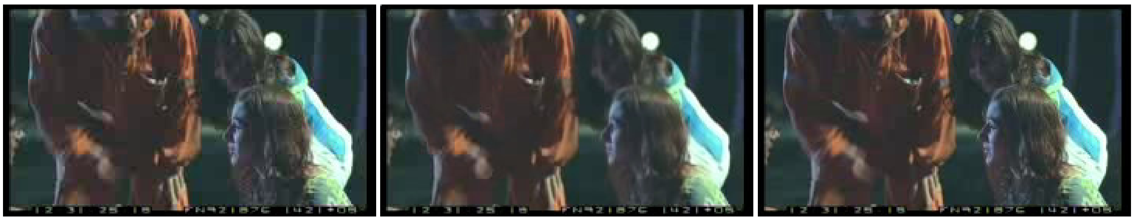

Fig. 15. *A frame of a film compressed with MPEG4 at 128kbps (left), denoised with AWL (middle), original frame compressed at 256kbps, without denoising (right.)*