Signal Processing 90 (2010) 2445–2455

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Nonlocal video denoising, simplification and inpainting using discrete regularization on graphs

Mahmoud Ghoniem *, Youssef Chahir, Abderrahim Elmoataz

GREYC, Campus 2, Boulevard du Marechal Juin, 14032 Caen, France

ARTICLE INFO

Article history:
Received 30 November 2008
Received in revised form
17 June 2009
Accepted 4 September 2009
Available online 12 September 2009

Keywords:
Video denoising
Video simplification
Video inpainting
Nonlocal discrete regularization
Weighted graphs

ABSTRACT

We present nonlocal algorithms for video denoising, simplification and inpainting based on a generic framework of discrete regularization on graphs. We express video denoising, simplification and inpainting problems using the same variational formulation. The main advantage of this framework is the unification of local and nonlocal approaches for these processing procedures. We take advantage of temporal and spatial redundancies in order to produce high quality results. In this paper, we consider a video sequence as a volume rather than a sequence of frames, and employ algorithms that do not require any motion estimation. For video inpainting, we unify geometric- and texture-synthesis-based approaches. To reduce the computational effort, we propose an optimized method that is faster than the nonlocal approach, while producing equally appealing results.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

The last decade has witnessed an overwhelming proliferation of video sequences due to technological progress. These sequences are often of poor quality, because of the acquisition system, the compression algorithm or the transmission process that may have been utilized. Noisy but *complete* video data are restored using the video denoising algorithms. The latter aim at enhancing the quality of the video by reducing noise as much as possible. In this context, the main challenge is to achieve the best trade-off between reducing noise and preserving significant structural elements. Nevertheless, corrupted videos with missing parts could be restored by inpainting the gaps using an adequate content consistently with the preserved parts of the video. Like painting retouching, inpainting must be hardly detectable.

Many methods have been proposed to denoise video sequences, and may be classified primarily based on the importance of motion estimation. Indeed, motion constitutes a fundamental challenge in video denoising. In this perspective, some methods involve a preliminary phase for motion estimation that is followed by a suitable denoising scheme, whereas other approaches incorporate the motion estimation into the denoising algorithm with or without smoothing constraints.

However, recent works have shown that resolving the motion estimation may not be advisable for the video denoising problem. Among these works, Protter and Elad [13] based their approach on the diffusion of a dictionary. Also, Buades et al. [4,3] worked on nonlocal filtering and demonstrated how the aperture problem can be taken advantage of. A variant of nonlocal filtering was presented by Boulanger et al. [2] using patches of variable sizes. Protter et al. extended the nonlocal-means in [12] for super-resolution reconstruction.

These nonlocal filters can be viewed as a regularization based on nonlocal functionals. Kindermann et al. [11] were the first to interpret nonlocal means and neighborhood filters as regularization based on nonlocal

* Corresponding author.
E-mail addresses: mahmoud.ghoniem@greyc.ensicaen.fr, mahmoud.ghoniem@gmail.com (M. Ghoniem), youssef.chahir@greyc.ensicaen.fr (Y. Chahir), abder.elmoataz@greyc.ensicaen.fr (A. Elmoataz).

0165-1684/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.sigpro.2009.09.004

functionals. Later, Gilboa and Osher [10] have proposed a quadratic functional of weighted differences for image regularization and semi-supervised segmentation. Elmoataz et al. [7] recently introduced a nonlocal discrete regularization framework, which is the discrete analogue of the continuous Euclidean nonlocal regularization functionals by Gilboa and Osher [10]. This method is applicable to image, mesh, manifold and data processing using weighted graphs of arbitrary topologies. Recently, we extended this framework to video denoising and simplification in [9,8].

Many methods have been proposed to inpaint video sequences. Inpainting algorithms fall under two main categories, namely, the geometric inpainting and the texture-based inpainting. The first category is based on partial differential equations (PDE). Bertalmio et al. proposed some preliminary work on frame-by-frame PDEs based video inpainting [1], whereby areas are filled by propagating information (linear structure) based on PDE from the outside of the masked region along level lines (isophotes). The PDE is applied spatially, and completes the video frame by frame. This approach does not take into account the temporal information that a video provides. The second group of inpainting algorithms is based on the texture synthesis. Wexler et al. [15] proposed a global optimization for space–time completion of holes in a video sequence. This method yields good results, but is computationally very expensive and over-smoothing is observed. The nonparametric sampling of Efros and Leung [6] is used in texture-based techniques. For a detailed survey of the state-of-the-art in video inpainting the more inquisitive reader is referred to [14].

In this paper, we propose a generic framework of discrete regularization on graphs that exploits and judiciously adapts a variational formulation for denoising, simplification and inpainting purposes on video media. The advantage of our framework lies in the unification of local and nonlocal approaches for these video processing procedures. For video inpainting, we unify a geometric approach and a texture-synthesis method. It is worth mentioning that our methods do not employ any motion estimation.

In Section 2, we present the principles governing regularization on weighted graphs. Thereafter, we state video denoising and simplification in terms of regularization on graphs, and detail our algorithm and its optimized variant. In Section 4, we present our video inpainting method as a novel extension of the denoising approach. In Section 5, we report and discuss empirical results for our proposed methods for video denoising, simplification and inpainting. We conclude this paper in Section 6 with a summary of our findings and directions for future research.

2. Regularization on weighted graphs

In this section, we introduce the derivative operators defined on a graph in order to formalize the concept of regularization on weighted graphs.

We adopt the nonlocal discrete regularization framework of Elmoataz et al. introduced in [7]. Upon defining the gradient and the p-Laplacian operators, the denoising and simplification problems are stated in terms of regularization on graphs.

2.1. Graph derivatives

Let $G=(V,E)$ be an undirected graph, where $V=\{v_1,v_2,\ldots,v_n\}$ is a finite set of vertices and $E\subseteq V\times V$ is a finite set of edges. Two vertices u and v are said to be adjacent if the edge $(u,v)\in E$. A graph is weighted if we associate to it a weight function $w: V\times V\to\mathbb{R}^+$, that satisfies the following conditions:

$$\forall (u,v)\in E,\quad w(u,v) = w(v,u), w(u,v)>0 \quad \text{if } u\neq v,\quad w(u,u) = 0 \text{ otherwise.}$$

We consider now the derivative operators on graphs we need for regularization. Let $\mathcal{H}(V)$ be a Hilbert space of real-valued functions on vertices. A function $f: V\to\mathbb{R}$ in $\mathcal{H}(V)$ assigns a vector f_v to each vertex v in V. The *local variation of the weighted gradient* operator $\|\nabla\|$ of a function $f\in\mathcal{H}(V)$ at a vertex v is defined by

$$\|\nabla f(v)\| = \sqrt{\sum_{u\sim v} w(u,v)(f(v)-f(u))^2}.$$

This can be viewed as a measure of the regularity of a function around a vertex.

The *weighted p-Laplace operator*, with $p\in]0,+\infty[$, at a vertex v is defined on $\mathcal{H}(V)$ by

$$(\Delta_p f)(v) = \frac{1}{p}\sum_{u\sim v}\gamma(u,v)(f(v)-f(u)),$$

where

$$\gamma(u,v) = w(u,v)(\|\nabla f(v)\|^{p-2} + \|\nabla f(u)\|^{p-2}).$$

2.2. p-Laplacian regularization on weighted graphs

Consider a function f^0 that could be an image, a video or any discrete data set. This function is defined over the vertices V of a weighted graph $G_w=(V,E,w)$ by $f^0: V\to\mathbb{R}$. f^0 is an observation of an original function f corrupted by a noise $n: f^0=f+n$.

The discrete regularization of $f^0\in\mathcal{H}(V)$ using the weighted p-Laplacian operator consists in seeking a function $f^*\in\mathcal{H}(V)$ that is not only smooth enough on G_w, but also close enough to f^0. It can be formalized by the minimization of two energy terms:

$$f^* = \min_{f\in\mathcal{H}(V)}\left\{\frac{1}{p}\sum_{v\in V}\|\nabla f(v)\|^p + \frac{\lambda}{2}\|f-f^0\|^2_{\mathcal{H}(V)}\right\}, \tag{1}$$

where $p\in]0,+\infty[$ is the smoothness degree, λ is the fidelity parameter, called the Lagrange multiplier, which specifies the trade-off between the two competing terms, and ∇f represents the weighted gradient of the function f over the graph. The solution of problem (1) leads to a family of nonlinear filters, parametrized by the weight function, the degree of smoothness, and the fidelity parameter.

The first energy in (1) is the smoothness term or regularizer, whereas the second is the fitting term. Both energy functions in E_p are strictly convex functions of f. In particular, by standard arguments in convex analysis, problem (1) has a unique solution, for $p\geq 1$, which satisfies

$$(\Delta_p f(v)) + \lambda(f(v)-f^0(v)) = 0,\quad \forall v\in V.$$

3. Video processing based on video regularization

We will use the previous regularization framework for video processing. To process video sequences, one can consider a video sequence as a simple sequence of independent frames and process each frame separately. Evidently, this would yield poor results from viewpoints of performance and processing quality. A video sequence mainly differs from a sequence of images in that the consecutive frames of a video are usually related due to the temporal redundancy from one frame to another. Therefore, the temporal dimension is an essential feature that should be integrated in the regularization algorithm itself. The extension to video sequences is based on the integration of time into the regularization process. We will take advantage of the high temporal redundancy of the data due to the high frame rates, which enhances the quality of our regularization. Hence, we develop a spatiotemporal regularization on weighted graphs.

3.1. Proposed algorithm

We consider the video sequence as a function f defined over the vertices of a weighted graph $G_{k_1,k_2,k_3}=(V,E,w)$, where $k_1,k_2,k_3\in\mathbb{N}^3$. A vertex v is defined by a triplet (i,j,t) where (i,j) indicates the spatial position of the vertex and t, which is a frame number, indicates the temporal position of the vertex within the video sequence. We denote by $u\sim v$ a vertex u that belongs to the neighborhood of v which is defined as follows:

$$N_{k_1,k_2,k_3}(v)=\{u=(i',j',t')\in V : \|i-i'\|\le k_1,\ \|j-j'\|\le k_2,\|t-t'\|\le k_3\}.$$

Similarly, we extend the definition of the patch to videos to obtain 3D patches. A patch around a vertex v is a box of size $r_x\times r_y\times r_t$, denoted by $B(v)$. Then, we associate to this patch a feature vector defined by

$$F(f^0,v)=f^0(u),\quad u\in B(v).$$

To be sure that each neighborhood contains more than one patch, the following relations must be respected: $k_1>\alpha*r_x, k_2>\alpha*r_y$ and $k_3\ge r_t$. Practically, we use $\alpha=3$. It is worth noting that r_x, r_y and r_t must be small enough compared with k_1, k_2 and k_3 for the nonlocal methods to ensure that the nonlocal neighborhood contains a significant number of patches.

The weight function w associated to G_{k_1,k_2,k_3} provides a measure of the distance between its vertices that can simply incorporate local, semi-local or nonlocal features according to the topology of the graph and the image. We consider the following two general weight functions:

$$w_L(u,v)=\exp\left(-\frac{|f(u)-f(v)^2|}{2\sigma_d^2}\right),$$

$$w_{NL}(u,v)=w_L(u,v)\cdot\exp\left(-\frac{\|F(f^0,u)-F(f^0,v)\|^2}{h^2}\right),$$

where σ_d^2 depends on the variations of $\|f(u)-f(v)\|$ over the graph. h can be estimated using the standard deviation depending on the variations of $\|F(f^0,u)-F(f^0,v)\|$ over the graph.

$w_L(u,v)$ is a measure of the difference between $f(u)$ and $f(v)$ values, and is used in the local approach of denoising. In addition to the difference between values, $w_{NL}(u,v)$ includes a similarity estimation of the compared features by measuring a $\mathcal{L}^2$ distance between the patches around u and v. It is the nonlocal approach.

To solve the regularization problem, we use the Gauss–Jacobi iterative algorithm presented in [7].

For all (u,v) in E:

$$\begin{cases} f^{(0)}=f^0,\\ \gamma^{(k)}(u,v)=w(u,v)(\|\nabla f^{(k)}(v)\|^{p-2}+\|\nabla f^{(k)}(u)\|^{p-2}),\\ f^{(k+1)}(v)=\dfrac{p\lambda f^0(v)+\sum_{u\sim v}\gamma^{(k)}(u,v)f^{(k)}(u)}{p\lambda+\sum_{u\sim v}\gamma^{(k)}(u,v)}, \end{cases} \tag{2}$$

where $\gamma^{(k)}$ is the function γ at the step k. The weights $w(u,v)$ are computed from f^0, or can be given as an input.

At each iteration, the new value $f^{(k+1)}$, at a vertex v, depends on two quantities, the original value $f^0(v)$ and a weighted average of the existing values in a neighborhood of v. This shows that the proposed filter, obtained by iterating, is a low-pass filter which can accommodate many graph structures and weight functions. For more details on the Gauss–Jacobi iterative algorithm, the reader is referred to [7]. Observe that for the specific parameters $p=2$, $\lambda=0$ and after one iteration we retrieve the nonlocal means solution of Buades et al. [4].

3.2. Algorithm optimization

The nonlocal method compares the patch around each vertex v with all the patches within its nonlocal neighborhood $N_{k_1,k_2,k_3}(v)$ and is observed to be slow. In the sequel, we reduced the processed data to $X\%$ and made an optimized nonlocal version. In our work, for each vertex v, we randomly select vertices from $N_{k_1,k_2,k_3}(v)$ to form the reduced nonlocal neighborhood of v. Hence, the optimized nonlocal algorithm is as follows:

Algorithm. For each vertex v of the graph:

Step 1: *Construct the reduced nonlocal neighborhood of v.* Select randomly $X\%$ of the vertices in $N_{k_1,k_2,k_3}(v)$ and out of the patch $B(v)$.

Step 2: *Compute the patch similarity* between $B(v)$ and $B(u)$ for all u in the reduced nonlocal neighborhood of v. All the patches $B(u)$ must be entirely included in $N_{k_1,k_2,k_3}(v)$.

Step 3: *Update the value* $f(v)$ according to (2)

The procedure described above is iterated N times. Alternatively, we utilize a termination criterion and the algorithm converges in fewer iterations. As a consequence, we accelerate the nonlocal method, and obtain equally attractive results as demonstrated in Section 5.

4. Video inpainting

In this section, we present the extension of our discrete regularization framework to video inpainting. The inpainting process consists in filling in the missing parts of the video with the most appropriate data in order to obtain harmonious and hardly detectable reconstructed zones. We propose to restore the missing data using the regularization approach.

This approach presents the advantage of completing the missing parts in harmony with the data in the neighborhood so that no sharp edges appear between the existing data and the reconstructed portion. Moreover, whereas some state-of-the-art methods involve an over-smoothing of the filled part, this shortcoming is precluded in our work by employing a nonlocal regularization method. Hence, we empirically observe that our results do not exhibit pronounced blur effects.

4.1. Formulation

Video inpainting is a specialization of the above-mentioned video regularization framework. First of all, in video inpainting, new values are computed for the missing parts of the video. In this context, there is no initial value for the missing parts and, hence, the fidelity parameter λ is set to 0. Let V_0 denote the nodes corresponding to the holes.

The discrete regularization formulation (1) can be written as

$$f^* = \min_{f\in\mathcal{H}(V)}\left\{\frac{1}{p}\sum_{v\in V}\|\nabla f(v)\|^p + \frac{\lambda(v)}{2}\|f - f^0\|^2_{\mathcal{H}(V)}\right\}, \tag{3}$$

where

$$\lambda(v) = \begin{cases} \lambda = \text{constant} & \text{if } v \in V/V_0, \\ 0 & \text{otherwise.} \end{cases}$$

For the missing parts, the discrete regularization formulation (3) reduces to

$$f^* = \min_{f\in\mathcal{H}(V)}\left\{\frac{1}{p}\sum_{v\in V}\|\nabla f(v)\|^p\right\}.$$

The system of equations (2) is specialized as follows: For all (u,v) in V_0,

$$\begin{cases} f^{(0)} = f^0, \\ \gamma^{(k)}(u,v) = w(u,v)(\|\nabla f^{(k)}(v)\|^{p-2} + \|\nabla f^{(k)}(u)\|^{p-2}), \\ f^{(k+1)}(v) = \dfrac{\sum_{u\sim v}\gamma^{(k)}(u,v)f^{(k)}(u)}{\sum_{u\sim v}\gamma^{(k)}(u,v)}, \end{cases} \tag{4}$$

where $\gamma^{(k)}$ is the function γ at the k th step. The weights $w(u,v)$ are typically computed from f^0, or could otherwise be explicitly given as an input.

The aforementioned framework unifies and subsumes several special techniques that are explored in the literature. In fact, by considering particular parameter values, we recover results that have been established in image processing. For $p=2$, w_{NL} and one iteration the nonlocal method is equivalent to the nonlocal means filter of Buades et al. [3] that has been adapted to inpainting by Wong et al. [16]. With $p=1$ and $w=1$, we obtain the local total variation (TV) inpainting of Chan and Shen [5]. The nonlocal TV inpainting corresponds to $p=1$ and w_{NL}. Our method could also be considered as an extension of Efros and Leung's work [6], where the k-nearest neighbors graph with $k=1$ is constructed and a patch distance between nodes is adopted.

4.2. Global description of our algorithm

The first step involves the construction of a mask corresponding to the missing parts of the video (V_0). Our method consists in filling that mask from its outer line to its center recursively. Thus, the missing regions are reconstructed by filling in a series of nested outlines. For a given pixel of the outer outline, we compute a new value using the known data in a volume centered in the curred pixel, which we refer to as the *search window*. Once the entire outer line is processed, it is removed from V_0.

This means that we do not update V_0 until an outline has been processed entirely. In other words, we do not include the computed value of a pixel in the estimation of the other pixels on the same level. Thus, we limit the risk of error propagation in our computations. Once an outline has been processed, it does not belong anymore to V_0 and is considered as known data that can be taken into account to process remaining holes. As the inpainting process progresses, the mask gets dynamically smaller and eventually becomes empty. At this point, all the holes are filled in.

The proposed algorithm is delineated as follows.

Algorithm. For each outer border o of the mask:
For each pixel p of o

- *Attribute a value to p*:
  - Construct the patch and the neighborhood of p.
  - Select the most similar patches to the patch around p within the search window.
  - Compute a value by regularizing the values of the centers of the selected patches. This value is attributed to p.
- *Update the mask*: Remove the processed outer border from the mask and process the new outer border.

Upon filling in all the holes of the video, the regularization process can be iteratively applied to the gaps that have been reconstructed by inpainting to obtain enhanced visual results.

4.3. Selection of the best candidates

This section details how to compute a new value for a missing node in harmony with the surrounding known data. To this end, we consider a patch around the node,

which may take various shapes depending on the mask shape. An illustration of different masks is given in Fig. 1.

A patch distance is defined to measure the similarity between our reference patch and the other patches of the search window. This metric provides a convenient way to describe the significant features of the vectors such as intensity or texture as relevant to the application at hand.

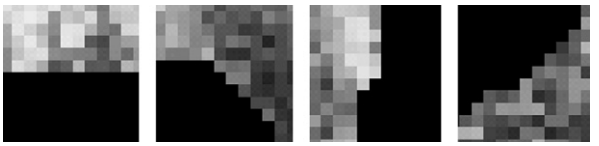

Fig. 1. Illustration of patches with different shapes.

The main challenge is to respect the overall appearance and preserve significant structural elements and details. Thus, the choice of the parameters of our process must be done according to the content to match within the video.

Indeed, it is worth noting that judicious parameters values, especially for the size of the search window and the size of the patch, are critical for the quality of the results, whereas inadequate values yield lower

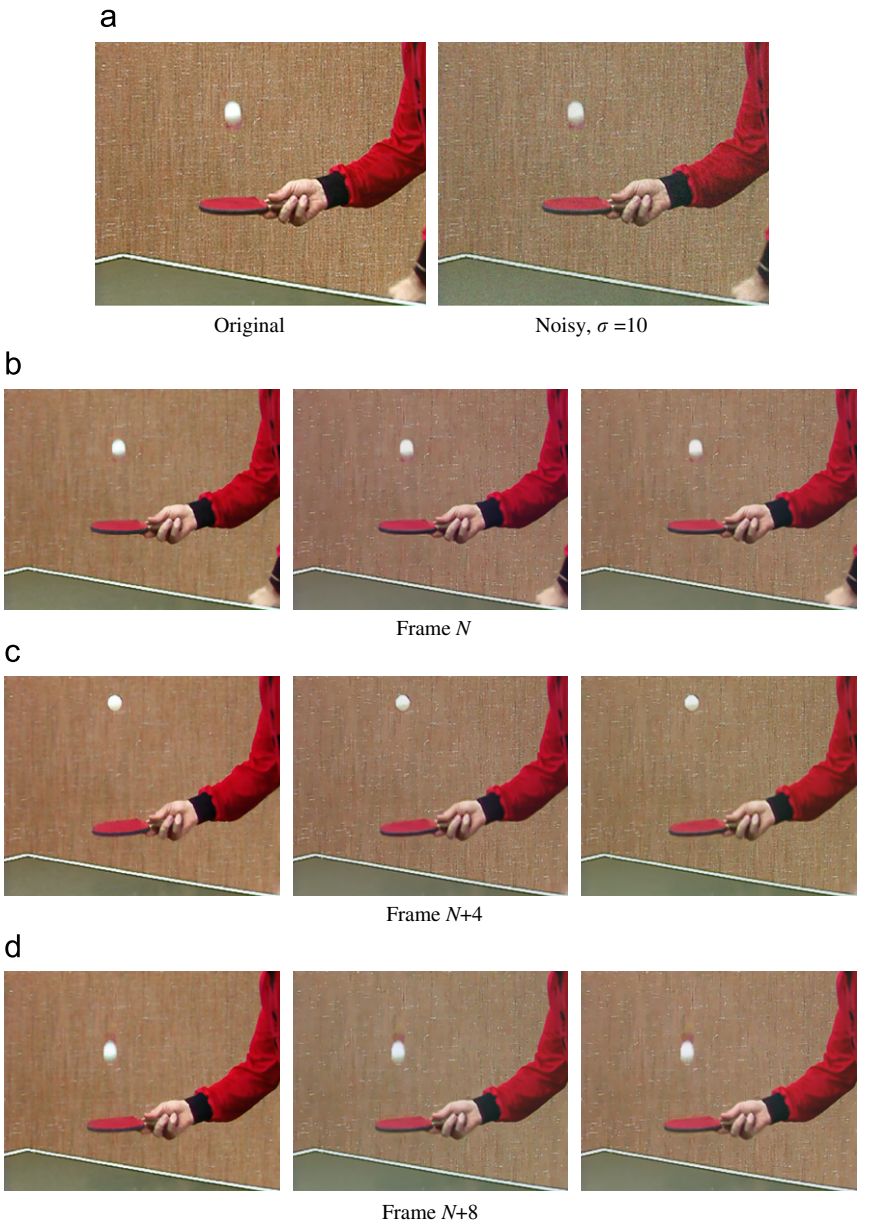

Fig. 2. Illustration of our three denoising methods with parameters $p = 1$, $\lambda = 0.5$ and $h = 30$ after three iterations: (a) original image; (b) noisy image; (c), (d) and (e): from left to right: local, nonlocal and optimized nonlocal results.

performances. To overcome this difficulty, we propose a multi-resolution variant of video inpainting. We introduce the use of patches of different sizes to best suit the homogeneity of the data zone under consideration. Large patches are adequate for homogeneous areas, whereas details and texture tend to require a variable size patch depending on the content represented by that data. The use of patches at different scales to select the closest candidates improves the robustness of our method. This variant has the advantage of automatically selecting the appropriate patch size, and is particularly appropriate for videos having zoom variations between frames.

Table 1
Comparison of the PSNR, expressed in dB, of noisy sequences (noise level $\sigma = 10$) denoised by our methods after one iteration.

Sequence	Size	Input	Local	Nonlocal	Optimized nonlocal
Flower	$180 \times 144 \times 126$	22.96	23.82	25.15	25.87
Tennis	$216 \times 172 \times 126$	24.68	27.12	28.65	28.18
Football	$180 \times 144 \times 105$	24.68	25.14	26.02	26.68
Mobile	$180 \times 144 \times 251$	21.24	24.31	27.08	26.22

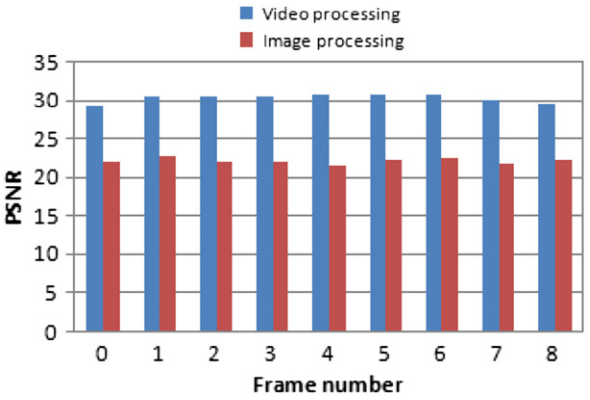

Fig. 3. PSNR comparison between video and image denoising results.

5. Experimental results

5.1. Video denoising

To test our algorithms, we considered the following parameters: $3 \times 3 \times 3$ patches, $7 \times 7 \times 3$ windows, $p = 1$, and $\lambda = 0.5$. We corrupted sequences with a synthetic zero-mean additive white Gaussian noise n having a variance σ^2.

As shown in Fig. 2, the nonlocal method provides the best visual result. Noise is highly reduced while thin structures such as textures and fine details are well preserved. The local method produces somewhat good results but some details are lost. Particularly, mobile objects such as the table tennis ball become very blurry with the local method whereas they are clear-cut in the results of the non local method. This test underlines the relevance of the nonlocal approach for denoising objects in motion. The optimized nonlocal method is very appealing as it preserves details and is faster than the nonlocal one.

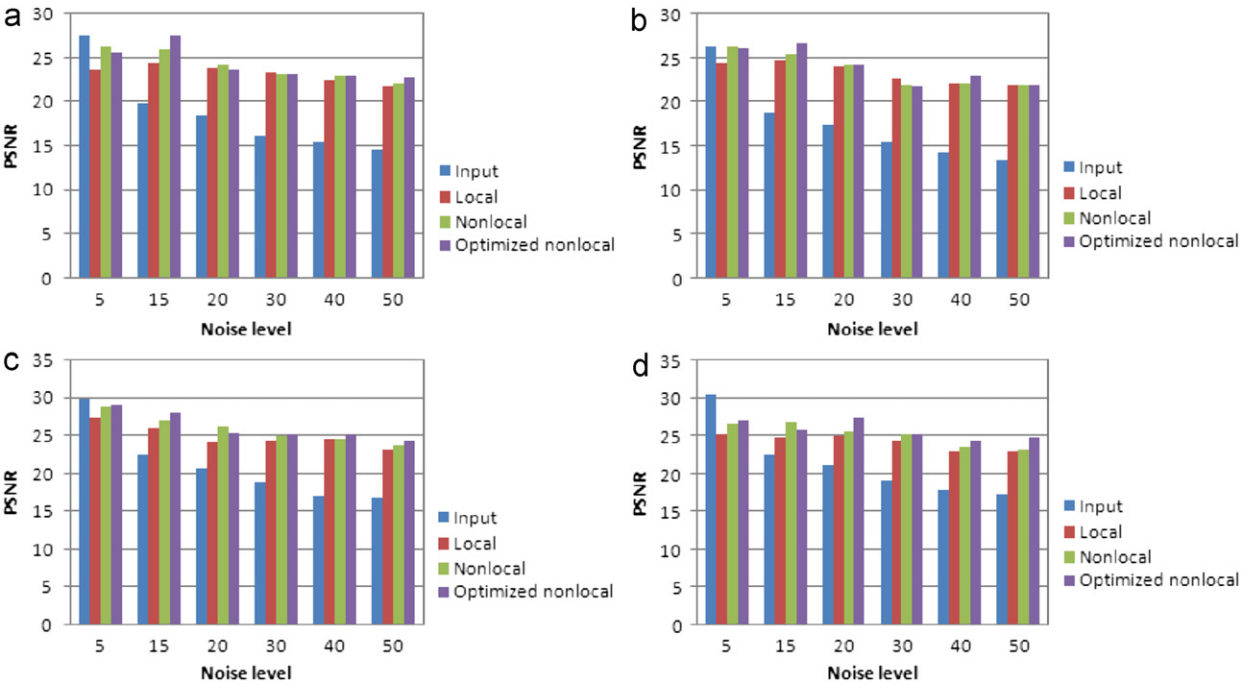

Fig. 4. PSNR comparison between our different denoising results and the input PSNR: (a) flower sequence; (b) mobile sequence; (c) tennis sequence; (d) football sequence.

The visual observations are further confirmed by the PSNR measures reported in Table 1. Although these results are slightly in favor of the nonlocal method, the proposed optimized variant produces very competitive results, using only 30% of the data, thereby achieving a faster computational performance.

Then we applied our video algorithm to the same video considered this time as a volume. We compared the 2D and 3D results. The PSNR of the video results are consistently higher than the PSNR of the 2D results as shown in Fig. 3. It is observed that video processing with the optimized nonlocal method provides a higher quality denoising than image processing.

This experiment evaluates and confirms the important contribution of temporal redundancy for video denoising.

It is also interesting to evaluate the efficiency of ourmethods for different noise levels. We run a bunch of tests on sequences corrupted by noises having σ from 5 to 50. The results are regrouped in Fig. 4. These results show how the input PSNR decreases significantly when the noise level increases, contrary to the results of our three methods. The higher the noise level is the biggest the difference is between the input PSNR and our results' PSNR. These measures also confirm that the optimized nonlocal method gives a similar PSNR and even a higher

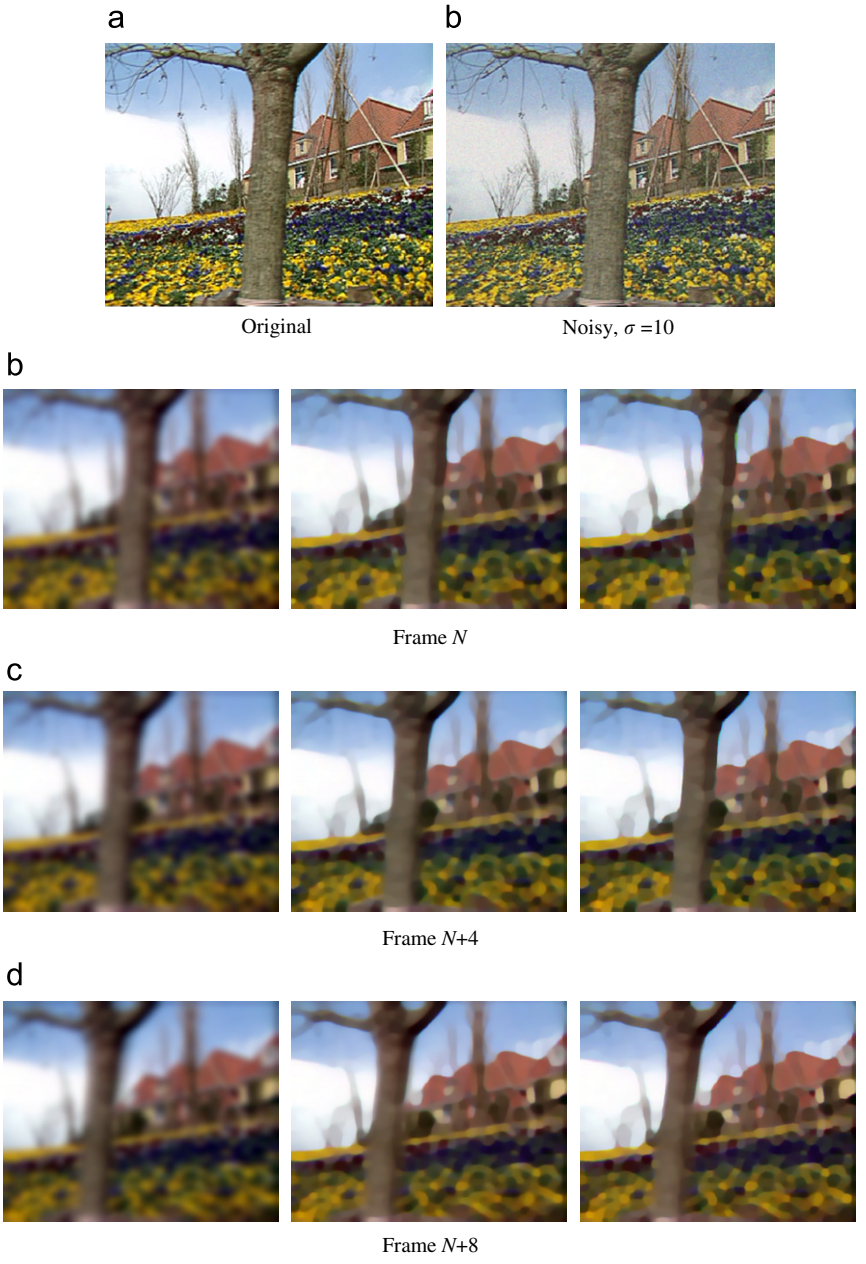

Fig. 5. Video simplification with the optimized nonlocal method, $\lambda = 0$ and $w = 1$ after five iterations. From left to right: $p = 2$, 0.5 and 0.1.

PSNR than the nonlocal one. In consequence, we consider that the optimized nonlocal method is a good trade-off between the fully nonlocal method and the local one from performance and quality viewpoints.

5.2. Video simplification

Video simplification can be interpreted as a particular case of our algorithm for $p<1$. To highlight the influence of p on the results, we used a constant weight function $w=1$. We obtain homogeneous partitions of the video content. In comparison with the result of $p=2$, we observe that, with $p=0.5$ and 0.1, similar regions join together and form bigger blocks and edges get sharper. Moreover, we obtain a coarser simplification of the video when p decreases (see Fig. 5). These results can be taken advantage of for video segmentation and visual object detection.

5.3. Video inpainting

We present now some tests that demonstrate the efficiency of our proposed inpainting procedure. To this

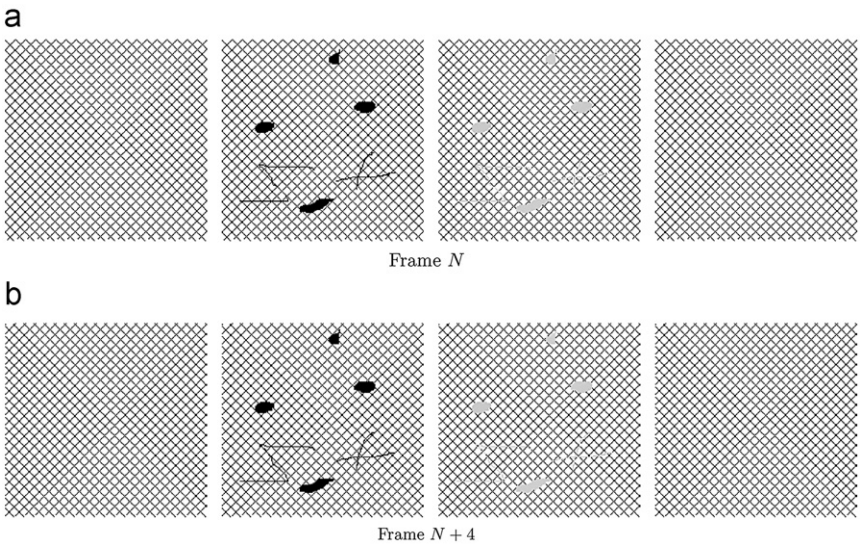

Fig. 6. Inpainting a synthetic video. Each frame of this video is a translation of the previous one. From left to right: the original synthetic image, the corrupted image, the result of local inpainting and the result of our nonlocal inpainting method.

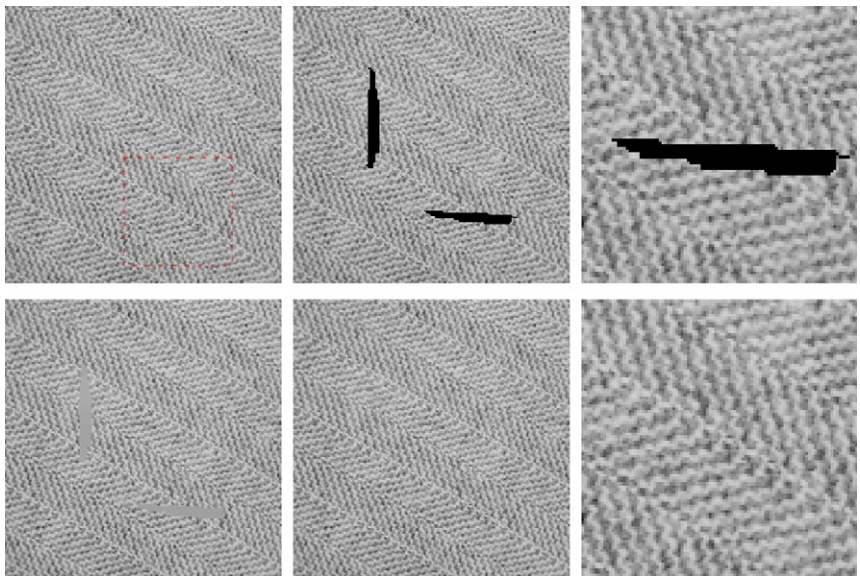

Fig. 7. Texture inpainting on weave image using a window 61 and patch 31. From left to right: on the first line, the original image, the corrupted image and a zoomed region of it, on the second line the result of a local inpainting, the result of our proposed nonlocal inpainting method and a zoomed region of it.

end, several media were selected, and the patch distance was determined based on the intensity.

In our first example, our algorithm was applied to a synthesis image. Fig. 6 indicates that the image was perfectly reconstructed using our nonlocal algorithm, whereas the local inpainting approach produced poor results. This test serves as a basic verification of the nonlocal algorithm.

Fig. 8. Inpainting on Barbara image using a window 21 and patch 5. From left to right, the original image, the corrupted image, the result of a local inpainting and our nonlocal inpainting result.

The result on a texture image is reported in Fig. 7. Observe that the reconstructed zones are not detectable and merge harmoniously with the uncorrupted data.

The results of the inpainting on complex images containing zones of variable homogeneity are also very encouraging. Fig. 8 reports the restoration of the reference image of Barbara. We purposely removed zones in a homogeneous region, textured zones, and zones of variable homogeneity to test the robustness of our method.

In Fig. 9, it is possible to observe the contribution of temporal redundancy in the inpainting process. We corrupted one frame in the well-known video corpus of Suzie. The results demonstrate the high efficiency of our nonlocal approach in coping with both texture and structures in the video. For instance, thin structures such as Suzie's hair or homogeneous zones like the background are equally well recovered whereas the local approach shows its limits.

To test our algorithm on text removal applications from videos, we added a static text to the foreman video and viewed the text as the holes to fill in. We can observe in Fig. 10 that the text is removed correctly and the inpainting yields very good results. Hence, our algorithm can be used for subtitle removal or any text removal application. Similarly, our method can be advantageously employed to remove objects from video sequences. In Fig. 11, we removed the ball from the video.

Our tests on several video corpus suggested that it is easier to remove several small zones than a unique large zone having the same surface. Indeed, in the case where small regions are scattered all over the video, many known parts exist and could be used to reconstruct the missing parts. On the contrary, when a compact large zone is missing, most of the candidates used to fill in

Fig. 9. On the first line, the original frames. On the second and third lines, from left to right, the corrupted frame, the local inpainting and the nonlocal video inpainting results using a window size of 21 and a patch size of 6.

Fig. 10. Text removal from a color video. From left to right, frames from the original video, the corrupted video and the result of our inpainting method.

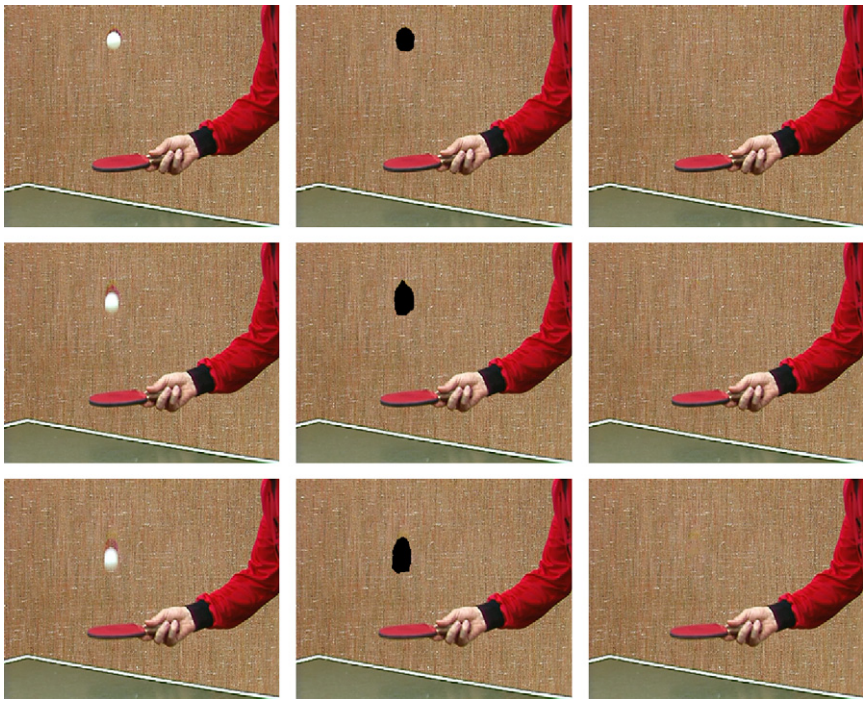

Fig. 11. Object removal from a color video. From left to right, frames from the original video, the corrupted video, the video without the tennis ball.

the inner part of the hole will be newly attributed value which results in a lower quality. One way to overcome this difficulty is to consider a large search window, but this induces an increased computational effort.

6. Conclusion and directions for future research

In this paper, a new algorithm has been introduced for the denoising, simplification and inpainting of video sequences. The algorithm does not require

any motion estimation and is based on a general framework of discrete regularization on weighted graphs that uses a variational formulation. Our framework presents the advantage of exploiting local and nonlocal regularities using patches to reduce noise while preserving significant features and details in the video. We benefit from temporal redundancies, which enhances the denoising results as demonstrated by our experiments. As the nonlocal method may become time-consuming, we reduced the amount of processed data (by 70% in our tests) without compromising the denoising quality. Our proposed video inpainting provides a novel method that unifies the geometric- and texture-based approaches.

Future research will investigate the acceleration of our methods by working on a pre-segmentation of the video. Our algorithm is flexible and covers a wide range of applications. Hence, we intend to use our video inpainting for video compression. Also, we are considering the video completion problem to improve our video inpainting method so as to fill in large space–time holes.

References

[1] M. Bertalmio, A.L. Bertozzi, G. Sapiro, Navier–Stokes, fluid dynamics, and image and video inpainting, I IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2001) 355–362.
[2] J. Boulanger, C. Kervrann, P. Bouthemy, Space–time adaptation for patch-based image sequence restoration, IEEE Trans. PAMI 29 (2007) 1096–1102.
[3] A. Buades, B. Coll, J.-M. Morel, Nonlocal image and movie denoising, Int. J. Comput. Vision 76 (2) (2008) 123–139.
[4] A. Buades, B. Coll, J.M. Morel, Denoising image sequences does not require motion estimation, in: IEEE Conference on Advanced Video and Signal Based Surveillance, 2005, AVSS 2005, 2005, pp. 70–74.
[5] T.F. Chan, J. Shen, Variational image inpainting, Commun. Pure Appl. Math. 58 (2005) 579–619.
[6] A. Efros, T. Leung, Texture synthesis by non-parametric sampling, in: IEEE International Conference on Computer Vision, Corfu, Greece, September 1999, pp. 1033–1038.
[7] A. Elmoataz, O. Lezoray, S. Bougleux, Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing, IEEE Trans. Image Process. 17 (7) (2008) 1047–1060.
[8] M. Ghoniem, Y. Chahir, A. Elmoataz, Video denoising and simplification via discrete regularization on graph, in: ACIVS, 2008, pp. 380–389.
[9] M. Ghoniem, Y. Chahir, A. Elmoataz, Video denoising via discrete regularization on graphs, in: ICPR08, 2008, pp. 1–4.
[10] G. Gilboa, S. Osher, Nonlocal linear image regularization and supervised segmentation, Multiscale Modeling Simulation 6 (2007) 595–630.
[11] S. Kindermann, S. Osher, P.W. Jones, Deblurring and denoising of images by nonlocal functionals, Multiscale Modeling Simulation 4 (2005) 1091–1115.
[12] M. Protter, M. Elad, H. Takeda, P. Milanfar, Generalizing the nonlocal-means to super-resolution reconstruction, IEEE Trans. Image Process 18 (1) (2009) 36–51.
[13] M. Protter, M. Elad, Sparse and redundant representations and motion-estimation-free algorithm for video denoising, in: Wavelets XII. Proceedings of the SPIE, vol. 6701, 2007, p. 43.
[14] M.V. Venkatesh, S. Cheung, J. Zhao, Efficient object-based video inpainting, Pattern Recognition Letters 30 (2009) 168–179.
[15] Y. Wexler, E. Shechtman, M. Irani, Space–time video completion, I IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2004) 120–127.
[16] A. Wong, J. Orchard, A nonlocal-means approach to exemplar-based inpainting, in: ICIP, 2008, pp. 2600–2603.