Alternate Structural-Textural Video Inpainting for Spot Defects Correction in Movies

A. Renaudeau[1(✉)], F. Lauze[2], F. Pierre[3], J.-F. Aujol[4], and J.-D. Durou[1]

[1] IRIT, UMR CNRS 5505, Université de Toulouse, France
[2] DIKU, University of Copenhagen, Denmark
[3] LORIA, UMR CNRS 7503, Université de Lorraine, INRIA projet Magrit, France
[4] Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251, F-33400 Talence, France
arthur.renaudeau@irit.fr

Abstract. We propose a new video inpainting model for movies restoration application. Our model combines structural reconstruction with a diffusion-based method and textural reconstruction with a patch-based method. Both proposed energies (one for each method) are alternatively minimized in order to preserve the overall structure while adding textural refinement. While the structural reconstruction is obtained jointly with optical flow computation with several proximal approaches, the textural reconstruction is processed by a variational non-local approach. Preliminary results on different Middlebury frames show quality improvement in the reconstruction.

1 Introduction

Video inpainting is a key issue for the movie industry, as it could help to automate the restoration of films that have suffered significant degradation (see Fig. 1), or the use of certain special effects that require the removal of elements for action scenes. Video inpainting, as every video processing, is increasingly being studied thanks to the power of processors and GPUs to perform large-scale calculations. Until now, video inpainting techniques have used separately diffusion-based methods with motion estimation, or patch-based methods with 3D patches to take into account temporal redundancy (similarity between consecutive frames), but without any explicit motion estimation this time. This motion can give a lot of information to recover data so it is a really good help for inpainting. However, in order to estimate motion, full data is needed and this is why this estimation must be processed at the same time as inpainting, which represents the main challenge.

In this paper, we aim to restore spot defects on previously digitized films. Each of these defects appears only in one frame and not in those located just before or just after. To eliminate them, our approach consists in combining a diffusion-based video inpainting model which jointly computes optical flow, with a patch-based model with 2D patches and shift maps to the temporally neighbouring images. With this approach, our model only needs the two adjacent frames to reconstruct the damaged area. While each model taken separately has drawbacks in terms of reconstruction quality, combining them both gives better results.

Fig. 1: Example of digitized frames from an old movie of the Cinémathèque de Toulouse with a defect in the central frame.

After reviewing related approaches in Section 2, we present our combined model in Section 3. Our numerical strategy for solving this variational problem, which is presented in Section 4, is based on alternating optimization of the diffusion-based and patch-based models. Preliminary experiments on Middlebury frames with added defects are conducted in Section 5, which confirm the interest of using both models together.

2 Related work

Inpainting is the name given to the technique of filling damaged or missing areas in an image. The term "inpainting" is only used from 2000 in [4], by analogy with the restoration process used in the field of art, after that of "disocclusion" in [21] in 1998. The first inpainting applications came from diffusion models for denoising, which date back to the early 1990s. This field of research has been very active in recent years, stimulated by many applications: removal of scratches or of text superimposed on an image, restoration of an altered image following a transmission, elimination of objects in an editing context for diminished reality.

Filling the area to be restored is an ill-posed inverse problem because there is not a single well-defined solution. It is therefore necessary to introduce a priori knowledge into the model. All existing methods are guided by the assumption that pixels located in known and missing parts of the image share the same statistical properties or geometric structures. This hypothesis is reflected in different local or global a priori assumptions, in order to obtain a restored image that is visually plausible. In diffusion-based inpainting, one wants to propagate the information contained in the pixels from the edge of the damaged area to the inside of this area. Total variation for inpainting was introduced in [11] to block the diffusion at the edges of the objects and recover piecewise constant data. The extension of diffusion-based inpainting to video started with [12] and [17,18] where motion is simultaneously estimated to fill the damaged area. From the well-known optical flow model of [16] with L^2 smooth regularization, [2] switched to L^1 norms to preserve discontinuities of the different motions, which was later solved using proximal algorithms in [23]. Very recently, [7] chose a complete TV-L^1 model to solve motion estimation and image reconstruction, using also proximal algorithms.

However, these diffusion-based models are limited because they cannot handle textures. This is why models based on full or partial patch copying have been developed (see more details in [8]) to keep details at high frequencies, starting with texture synthesis in [15] and then local patch-based inpainting (patches are only looked for in a neighbourhood of the defect area) in [13] with priority for filling based on the magnitude of the spatial gradient at the edges of the area to be filled. Finally, recent methods consider a mixture of patches following a spatial non-local search as in [1]. This patch-based approach for inpainting was also extended to videos in [22] and [19] with 3D patches to include some temporal similarity in patch comparisons. If these models yield better results in the recovery of texture, they are however highly dependent on the initial filling of the area, so as not to remain blocked in a local minimum for the solution, which usually fails at reconstructing regular structures. Moreover, considering the patch size is also an important criterion in order to recover textures with different statistical properties.

In the image inpainting context, the idea of mixing both approaches has already been developed in [14] where diffusion and texture filling are sequentially processed, in [6] where the image is decomposed into cartoon and texture before being filled separately, and in [9] where texture is filled guided by the level lines. In the video denoising context, [5] uses patches combined by the computation of a structural optical flow of [23]. Our aim in this paper is also to get the best of both worlds by combining diffusion to recover structure with patches for texture, with also diffusion-based and patch-based approaches for motion estimation.

3 Statement of the problem

Let us define a sequence of 3 successive color frames $\{u_b, u, u_f\}$ as functions $\Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ with bounded variation, where u_b is the backward frame, u is the current frame containing the defect to be inpainted, and u_f is the forward frame. This defect area is defined by $O \subset \Omega$. The problem to be solved is as follows:

$$\left\{v_b^*, \Gamma_b^*, u^*, v_f^*, \Gamma_f^*\right\} = \underset{v_b, \Gamma_b, u, v_f, \Gamma_f}{\operatorname{argmin}} \left\{E_S(v_b, u, v_f) + E_T(\Gamma_b, u, \Gamma_f)\right\} \quad (1)$$

where E_S represents the energy for structural reconstruction, minimized channel by channel (or using the luminance channel for the motion estimation), and E_T the energy for textural reconstruction, minimized using color frames directly. The different variables are explained in the following subsections.

3.1 Structural reconstruction energy

The model chosen for E_S is based on the works of [18] and [7], adding a symmetry between the optical flows (reminding u is here only one channel):

$$\begin{aligned} E_S(v_b, u, v_f) = \mu \int_\Omega |\nabla u(x)| \, \mathrm{d}x + \lambda \int_\Omega |Jv_b(x)| \, \mathrm{d}x + \lambda \int_\Omega \left|Jv_f(x)\right| \, \mathrm{d}x \\ + \int_\Omega |u_b(x + v_b(x)) - u(x)| \, \mathrm{d}x + \int_\Omega \left|u_f(x + v_f(x)) - u(x)\right| \, \mathrm{d}x \end{aligned} \quad (2)$$

under the constraint $u = u^0$ over $O^c = \Omega \backslash O$ to preserve the healthy part of the frame. The terms containing the motion field $v_{_b} : \Omega \to \mathbb{R}^2$ (respectively $v_{_f}$) represent the L^1 regularized optical flow constraint, proposed by [23], between the current frame u and the backward frame $u_{_b}$ (respectively the forward frame $u_{_f}$), but using Jacobian matrices, as a rewriting of the formula in [10]. Every integral contains a discrete norm $|.|$ defined as $|M| = \sqrt{\sum_{i,j} m_{_{i,j}}^2}$, which means either an absolute value, a vector norm or a Frobenius norm depending on the case. The parameters λ and μ are used to define the trade-off between data fitting and regularization.

3.2 Textural reconstruction energy

The second energy E_T is an extension to video of the work of [1], using directly the color frames (not channel by channel as for the structural energy). Here the search for optimal patches is no longer carried out in a spatial neighbourhood around the defect, but in a temporal neighbourhood (in the previous frame $u_{_b}$ and the next frame $u_{_f}$). While in [22] and [19] the 3D patch search is not limited in time distance, here we focus only on 2D patches in the backward and forward frames. In the current frame u, the central pixel x of the patch $p_u(x)$ concerned by the search of the optimal patch in the neighbour frames $u_{_b}$ and $u_{_f}$ is in the area $\widetilde{O}$ which is O expanded by half a patch width, in order to propagate patches containing sufficient healthy data:

$$\begin{aligned} E_T(\Gamma_{_b}, u, \Gamma_{_f}) = &\int_{\tilde{O}} \omega(x)\, \varepsilon\Big[p_{u_{_b}}(\Gamma_{_b}(x)) - p_u(x)\Big]\, \mathrm{d}x \\ &+ \int_{\tilde{O}} (1-\omega(x))\, \varepsilon\Big[p_{u_{_f}}(\Gamma_{_f}(x)) - p_u(x)\Big]\, \mathrm{d}x \end{aligned} \tag{3}$$

where $\Gamma_{_b}$ and $\Gamma_{_f}$ are shift maps, respectively, from u to $u_{_b}$ and from u to $u_{_f}$, $\omega(x) \in [0,1]$ is a weight between the two possible reconstructions of u from forward or backward frames (see **Section 4.3**), and ε represents the chosen distance between patches. Here in (4), ε is a convolution between the squared difference of the patches $(p_{u_{_b}}(\Gamma_{_b}(x)) - p_u(x))^2$ and a Gaussian kernel g_a of standard deviation a for a non-local means reconstruction:

$$\varepsilon\Big[p_{u_{_b}}(\Gamma_{_b}(x)) - p_u(x)\Big] = \int_{\Omega_p} g_a(x_p)\, \left[u_{_b}(\Gamma_{_b}(x) - x_p) - u(x - x_p)\right]^2 \mathrm{d}x_p \tag{4}$$

where $x_p \in \Omega_p$ denotes the coordinates of a pixel inside the patch $p_u(x)$ relative to its center x. Minimizing E_S and E_T at the same time is a very complex problem with no proof of existence and uniqueness of a solution for (1), but one only wishes to obtain an approximate numerical solution by minimizing E_S and E_T alternatively, using the result u of the minimization of one to initialize the other one.

4 Optimization

Applying inpainting to large defects or estimating motions requires a coarse-to-fine framework. In a video context, it is even more important to follow this strategy in order to initialize every variable correctly. The idea here is to downsample enough the frames to consider that motions are small enough between them. At such a resolution (the level $L \to L_{\max}$), considering Gaussian filtering to eliminate high frequencies in the downsampling step, the structural reconstruction works well whereas the textural one is not efficient. On the other hand, at higher resolution ($L \to 0$), we want to put more emphasis on texture. This is why our algorithm can choose a maximum resolution level $L_{\max}^{\text{texture}}$ to start texture reconstruction and a minimum resolution level $L_{\min}^{\text{structure}}$ to stop structural reconstruction, with $L_{\max}^{\text{texture}} \geq L_{\min}^{\text{structure}} - 1$ to ensure that at least one of the reconstructions is applied at each resolution. Consequently, at a given resolution level $L > 0$, our algorithm applies only one reconstruction or both in a row:

Algorithm 1 - Reconstruction at resolution levels L and $L-1$

if $L \geq L_{\min}^{\text{structure}}$ **then**
  $v_b^*, u^*, v_f^* \leftarrow \underset{v_b, u, v_f}{\operatorname{argmin}} \left\{ E_S\left(v_b, u, v_f\right) \right\}$
  $v_b, v_f \leftarrow \text{upsampling}(v_b^*, v_f^*)$
  $u \leftarrow u^*$
end if

if $L \leq L_{\max}^{\text{texture}}$ **then**
  $\Gamma_b^*, \Gamma_f^* \leftarrow \underset{\Gamma_b, \Gamma_f}{\operatorname{argmin}} \left\{ E_T\left(\Gamma_b, u, \Gamma_f\right) \right\}$
  $\Gamma_b, u, \Gamma_f \leftarrow \text{upsampling}(\Gamma_b^*, u, \Gamma_f^*)$
  $u^* \leftarrow \underset{u}{\operatorname{argmin}} \left\{ E_T\left(\Gamma_b, u, \Gamma_f\right) \right\}$
  $u \leftarrow u^*$
else
  $u \leftarrow \text{upsampling}(u)$
end if

Notice that shift maps upsampling is carried out with the nearest neighbour interpolation, while the bicubic one is used for the other upsamplings. The different minimizations of u, v_b, v_f, Γ_b and Γ_f in **Algorithm 1** are explained below.

4.1 Motion estimation

In order to minimize E_S with respect to the motion vector v_b, we proceed as in [23] by linearizing inside the two absolute differences in (2). However, this linearization is only possible in the case of small displacements. This is why a constant motion vector v_b^0 is introduced, close to v_b, around which the latter is estimated, to get:

$$\begin{aligned} E_S(v_b, u, v_f) = \mu \int_\Omega |\nabla u(x)| \, \mathrm{d}x + \lambda \int_\Omega |Jv_b(x)| \, \mathrm{d}x + \lambda \int_\Omega |Jv_f(x)| \, \mathrm{d}x \\ + \int_\Omega |\nabla u_b(x + v_b^0(x)) \cdot [v_b - v_b^0](x) + u_b(x + v_b^0(x)) - u(x)| \, \mathrm{d}x \\ + \int_\Omega |\nabla u_f(x + v_f^0(x)) \cdot [v_f - v_f^0](x) + u_f(x + v_f^0(x)) - u(x)| \, \mathrm{d}x \end{aligned} \tag{5}$$

where $\nabla u_b(x+v_b^0(x))\cdot[v_b-v_b^0](x)+u_b(x+v_b^0(x))-u(x)$ will appear as $\rho(u,v_b,u_b)$ afterwards (same goes for v_f). Minimizing with respect to the motion vector v_b leads to the form:

$$v_b^* = \underset{v_b}{\operatorname{argmin}} \max_y \int_\Omega |\rho(u,v_b,u_b)|\,\mathrm{d}x + \langle\, Jv_b \,|\, y \,\rangle - \iota_{B^\infty}\left(\frac{y}{\lambda}\right) \tag{6}$$

introducing the dual variable of v_b, $y:\Omega\to\mathbb{R}^{2\times 2}$. This convex problem can be solved by the primal-dual algorithm of [10], noticing that $Jv_b=[\nabla v_{b,1},\nabla v_{b,2}]^\top$ and so we get the adjoint operator $J^*y=-[\mathrm{div}([y_{1,1},y_{1,2}]^\top),\mathrm{div}([y_{2,1},y_{2,2}]^\top)]^\top$. Whereas the proximal operator associated to y is a projection onto the L^∞-norm ball, the proximal operator associated to v_b is a soft thresholding (see [23] for details):

$$\begin{cases} y^{(n+1)} \leftarrow \mathrm{prox}_{\lambda\sigma\iota_{B^\infty}}\left(y^{(n)}+\sigma J\bar{v}_b^{(n)}\right) \\ v_b^{(n+1)} \leftarrow \mathrm{prox}_{\tau\rho(u,-,u_b)}\left(v_b^{(n)}-\tau J^*y^{(n+1)}\right) \\ \bar{v}_b^{(n+1)} \leftarrow v_b^{(n+1)}+\theta\left(v_b^{(n+1)}-v_b^{(n)}\right) \end{cases} \tag{7}$$

where $\sigma,\tau>0$ are time steps and $\theta\in[0,1]$. The minimization of E_S with respect to v_f is carried out in a similar way.

4.2 Structural reconstruction

After motion has been estimated, the inpainting process is obtained by minimizing:

$$\begin{aligned} u^* = \underset{u}{\operatorname{argmin}} \quad & \int_\Omega |u_b(x+v_b(x))-u(x)|\,\mathrm{d}x \\ & + \int_\Omega |u_f(x+v_f(x))-u(x)|\,\mathrm{d}x + \mu\int_\Omega |\nabla u(x)|\,\mathrm{d}x \end{aligned} \tag{8}$$

In order to rewrite the convex problem (8) with dual variables, the time variable must be clarified with respect to u, v_b and v_f. Indeed, taking 1 as the time step between two frames, and t as the current time for u, then u_b and u_f take the form:

$$\begin{aligned} u_b(x+v_b(x)) &= u(x+v_b(x,t),t-1) = u\left(\varphi_b(x,t)\right) \\ u_f(x+v_f(x)) &= u(x+v_f(x,t),t+1) = u\left(\varphi_f(x,t)\right) \end{aligned} \tag{9}$$

with φ_b and φ_f two transformations, which are similar to shift maps, with the hypothesis that $\varphi_b\circ\varphi_f=\varphi_f\circ\varphi_b=I_d$ almost everywhere. With these new notations, (8) can be rewritten as:

$$u^* = \underset{u}{\operatorname{argmin}} \max_z \left\langle \begin{matrix} u\circ\varphi_b-u \\ u\circ\varphi_f-u \\ \nabla u \end{matrix} \middle| z \right\rangle - \iota_{B^\infty}(z_1) - \iota_{B^\infty}(z_2) - \iota_{B^\infty}\left(\frac{1}{\mu}\begin{bmatrix} z_3 \\ z_4 \end{bmatrix}\right) \tag{10}$$

introducing the dual variable of u, $z:\Omega\to\mathbb{R}^4$. Noting K the operator with respect to u in the inner product, it leads to the adjoint operator K^* as in [18]:

$$K^*z = \det\left(J\varphi_f\right) z_1\circ\varphi_f - z_1 + \det\left(J\varphi_b\right) z_2\circ\varphi_b - z_2 - \mathrm{div}([z_3,z_4]^\top) \tag{11}$$

Minimizing (10) can also be carried out, using the primal-dual algorithm of [10]:

$$\begin{cases} z_1^{(n+1)} \leftarrow \text{prox}_{\sigma \iota_{B^\infty}}\left(z_1^{(n)} + \sigma\left(u \circ \varphi_b - \bar{u}^{(n)}\right)\right) \\ z_2^{(n+1)} \leftarrow \text{prox}_{\sigma \iota_{B^\infty}}\left(z_2^{(n)} + \sigma\left(u \circ \varphi_f - \bar{u}^{(n)}\right)\right) \\ \begin{bmatrix} z_3^{(n+1)} \\ z_4^{(n+1)} \end{bmatrix} \leftarrow \text{prox}_{\mu\, \sigma' \iota_{B^\infty}}\left(\begin{bmatrix} z_3^{(n)} \\ z_4^{(n)} \end{bmatrix} + \sigma' \nabla \bar{u}^{(n)}\right) \\ u^{(n+1)} \leftarrow u^{(n)} - \tau K^* z^{(n+1)} \\ \bar{u}^{(n+1)} \leftarrow u^{(n+1)} + \theta\left(u^{(n+1)} - u^{(n)}\right) \end{cases} \tag{12}$$

where $\sigma, \sigma', \tau > 0$ are time steps and $\theta \in [0, 1]$.

To minimize an energy similar to (2), [7] repeats alternating minimizations between inpainting u and estimating a unique motion vector field v until convergence. In our case, instead of doing such a thing, it was decided to minimize the three variables (v_b, u, v_f) together, by applying successive proximal steps on each variable. The three descents in v_b, v_f and u give thus better results in the reconstruction than the first option for an equivalent computation time.

4.3 Shift maps estimation and textural reconstruction

The textural reconstruction is based on the work of [1], where the shift maps Γ_b and Γ_f are estimated using the PatchMatch algorithm of [3], with a L^2-distance between patches. Consequently, for Γ_b (respectively Γ_f), $\forall x \in \widetilde{O}$:

$$\Gamma_b(x) = \underset{x_b \in \Omega}{\text{argmin}} \int_{\Omega_p} g_a(x_p)\, \left[u_b(x_b - x_p) - u(x - x_p)\right]^2 \mathrm{d}x_p \tag{13}$$

For each temporal neighbour frame u_b and u_f, each associated part of (3) can be rewritten, without considering the weight ω for now, as the extreme case of choosing only the nearest neighbour patch for every pixel in the defect area (see [1] for details), which introduces a Dirac δ:

$$E_T^b(u, \Gamma_b) = \int_{\tilde{O}} \int_{\Omega} \delta(\Gamma_b(x) - x_b)\, \varepsilon[p_{u_b}(x_b) - p_u(x)]\, \mathrm{d}x_b\, \mathrm{d}x \tag{14}$$

With the changes of variables $x := x - x_p$ and $x_b := x_b - x_p$ which operate two translations, we obtain from (4) and (14):

$$E_T^b(u, \Gamma_b) = \int_O \int_{\Omega} m(x, x_b)\, \left[u_b(x_b) - u(x)\right]^2 \mathrm{d}x_b\, \mathrm{d}x \tag{15}$$

with:

$$m(x, x_b) = \int_{\Omega_p} g_a(x_p)\ \delta(\Gamma_b(x + x_p) - (x_b + x_p))\, \mathrm{d}x_p \tag{16}$$

whose integral over Ω is equal to 1 since g_a is assumed normalized. By expanding the squared difference in (15), u is also the minimizer of the following energy, which is equal to $E_T^b(u, \Gamma_b)$ up to a constant:

$$\tilde{E}_T^b(u, \Gamma_b) = \int_O \left[u(x) - \int_\Omega m(x, x_b)\, u_b(x_b)\, \mathrm{d}x_b \right]^2 \mathrm{d}x \tag{17}$$

which directly leads to the non-local means solution defined $\forall x \in O$:

$$u(x) = \int_\Omega m(x, x_b)\, u_b(x_b)\, \mathrm{d}x_b = \int_{\Omega_p} g_a(x_p)\, u_b(\Gamma_b(x + x_p) - x_p))\, \mathrm{d}x_p \tag{18}$$

This is the result for only one of the two neighbour frames. To take into account both frames, some weighted means are introduced with the weights $\omega(x)$ and $(1 - \omega(x))$. Choosing half of both results ($\omega(x) = 0.5$, named T_H afterwards) will cause some blur. However, taking only the best ($\omega(x) \in \{0, 1\}$, named T_B) will cause spatial artifacts. In this case, the choice of the weight 0 or 1 is given after both shift maps Γ_b and Γ_f having been estimated. Then, for each pixel to recover, the weight $\omega(x)$ is equal to 1 if the final distance $\varepsilon \left[p_{u_b}(\Gamma_b(x)) - p_u(x) \right]$ from the backward frame is smaller than the one from the forward frame, i.e. $\varepsilon \left[p_{u_f}(\Gamma_f(x)) - p_u(x) \right]$, and is equals to 0 otherwise. An intermediary solution can be to take a weighted mean (named T_W^α), using the following ratio containing both previous distances ε:

$$\omega(x) = \frac{\varepsilon \left[p_{u_f}(\Gamma_f(x)) - p_u(x) \right]^\alpha}{\varepsilon \left[p_{u_f}(\Gamma_f(x)) - p_u(x) \right]^\alpha + \varepsilon \left[p_{u_b}(\Gamma_b(x)) - p_u(x) \right]^\alpha} \tag{19}$$

where the power $\alpha \in [0, +\infty)$. One can notice that this formula can generalize the two first choices as follows:

$$\lim_{\alpha \to 0} T_W^\alpha = T_H \quad \text{and} \quad \lim_{\alpha \to +\infty} T_W^\alpha = T_B. \tag{20}$$

In the next section, the use of T_W without specifying α means that α is equal to 1, which is the case of a classical ratio between the distance from the forward frame and the sum of both distances.

5 Experiments

Our algorithm is implemented on Matlab and C, using the maximum level $L_{\max} = 10$ in multiresolution pyramid with a factor of $\sqrt{2}$. In order to test the algorithm, three consecutive Middlebury frames were used where a defect was artificially put in the central frame (see Fig. 2). The videos of the complete Middlebury sequences of 8 frames with the defect, and the reconstructions are available for downloading at the following link: *SSVM VideoInpainting*.

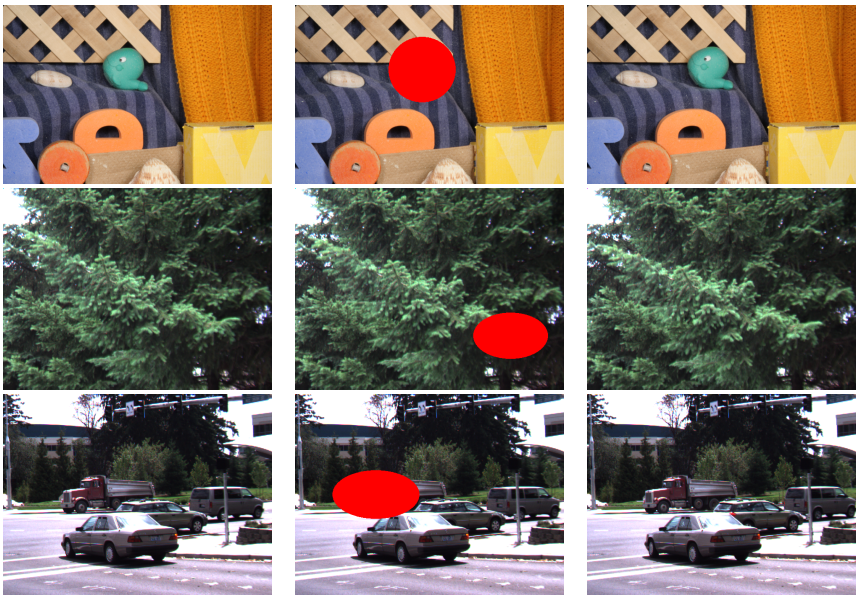
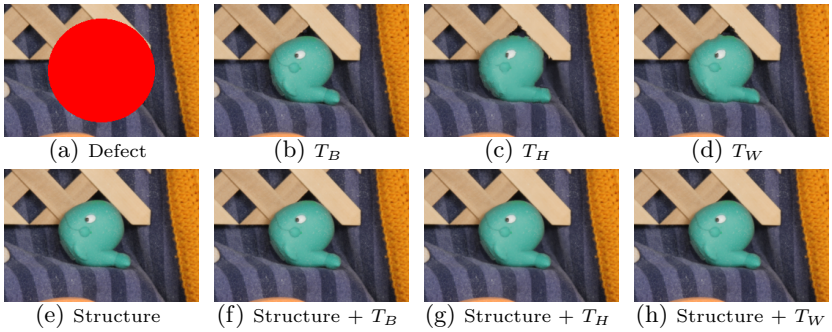


Fig. 2: Frames 8 to 10 from "RubberWhale" (top), "Evergreen" (middle) and "Dumptruck" (bottom) sequences with a defect highlighted in red in the central frame.

5.1 Qualitative results

Concerning the "RubberWhale" frames, results in Fig. 3 show that the textural methods (3b to 3d) are inadequate to reconstruct the puppet and the backward properly, whatever the chosen patch. On the other hand, the structural reconstruction (3e) is already good. However, our algorithm still gives some little improvement (3f to 3h), in particular at the top of the head of the puppet where there is false color with the structure, due to the channel separation to inpaint, and at the corner of the wooden bar at the back which is less rounded.

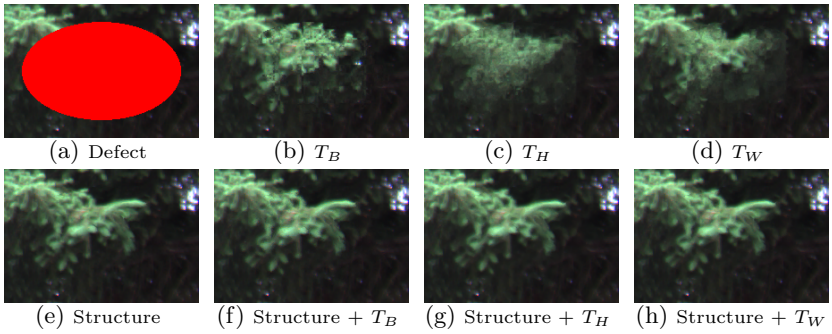


(a) Defect (b) T_B (c) T_H (d) T_W

(e) Structure (f) Structure + T_B (g) Structure + T_H (h) Structure + T_W

Fig. 3: "RubberWhale" reconstructions (zooms on the defect area).

In order to show more improvements, the two next sequences represent realistic scenes with more texture and larger motions. In the case of "Evergreen" frames for instance, results in Fig. 4 show that textural methods (4b to 4d) cannot manage to reconstruct the tree branches properly. The best patch approach (4b) leads to artifacts wheareas the mean approaches (4c and 4d) operates an averaging, as expected, which results in a lack of texture. The structural reconstruction (4e) is better but still a little blurry because of the diffusivity and some branches seems a little bit stretched out. On the other hand, the combination of both reconstructions (4f to 4h) succeeds in adding texture to structure.

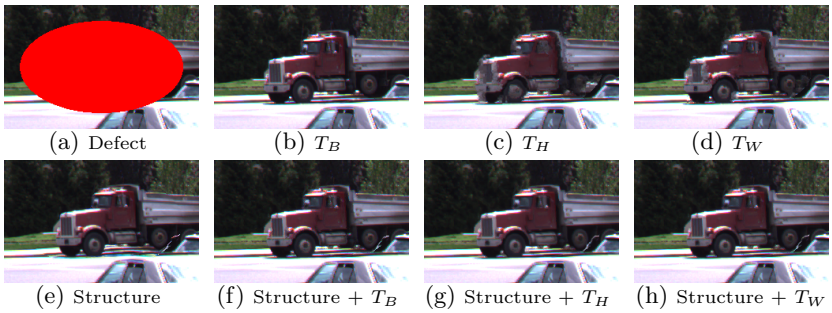


(a) Defect (b) T_B (c) T_H (d) T_W

(e) Structure (f) Structure + T_B (g) Structure + T_H (h) Structure + T_W

Fig. 4: "Evergreen" reconstructions (zooms on the defect area).

The results for "Dumptruck" frames in Fig. 5 show that the structural reconstruction (5e) leads to a deformation of the truck, which seems to oscillate vertically in the video. Moreover, there is also some ghosting effect behind the car that goes to the right. Textural reconstructions using means (5c and 5d) lead, as expected, to blurry reconstructions of the truck. Even if using the best patch from both adjacent frames (5b), whose result seems to be really good on the static reconstructed frame, the video shows that the algorithm chooses the closest frame in terms of patch distance and stays locked, this is why a lack of motion appears in the defect area in the video. Using our algorithm (5f to 5h), we obtain better results with the textural refinement, which permits to keep a good optical flow. Only the ghosting effect remains, due to the large displacement of the car.



(a) Defect (b) T_B (c) T_H (d) T_W

(e) Structure (f) Structure + T_B (g) Structure + T_H (h) Structure + T_W

Fig. 5: "Dumptruck" reconstructions (zooms on the defect area).

5.2 Quantitative results

Regarding the PNSR and SSIM quality metrics for the "RubberWhale" frame reconstructions in Table 1 (up), it leads to the same interpretation: the structural reconstruction performs as well as our algorithm. For the "Evergreen" frame reconstructions, the first visual interpretations are also validated by the quality metrics in Table 1 (middle) where structural reconstruction performs better than textural reconstruction, and our algorithm gives the best results with a certain gap. Moreover, choosing any of the three textural reconstructions with the structural one has no longer a real impact on the result. Concerning the "Dumptruck" frame reconstructions, by looking at the quality metrics in Table 1 (bottom), our algorithm leads to a significant improvement compared to the separate structural and textural reconstructions.

RubberWhale	-	Textural reconstructions		
		T_B	T_H	T_W
-	16.34 - 0.915	38.46 - 0.994	41.12 - 0.996	39.92 - 0.995
Structural reconstruction	**48.49 - 0.999**	46.99 - 0.999	**48.81 - 0.999**	**48.59 - 0.999**

Evergreen	-	Textural reconstructions		
		T_B	T_H	T_W
-	18.66 - 0.943	30.94 - 0.978	33.37 - 0.980	33.56 - 0.982
Structural reconstruction	37.24 - 0.991	**40.37 - 0.994**	**40.47 - 0.994**	**40.50 - 0.994**

Dumptruck	-	Textural reconstructions		
		T_B	T_H	T_W
-	17.53 - 0.937	33.16 - 0.987	34.35 - 0.989	34.31 - 0.988
Structural reconstruction	27.76 - 0.975	**38.11 - 0.994**	**38.87 - 0.994**	**38.58 - 0.994**

Table 1: PSNR - SSIM for the different reconstructions of the "RubberWhale" frame (top), the "Evergreen" frame (middle) and the "Dumptruck" frame (bottom).

6 Conclusion and perspectives

In this paper, we have shown that combining structural reconstruction based on diffusion approaches and textural reconstruction leads to better results in terms of visual and metrics quality. To go further, it would be interesting to refine the optical flow model using total generalized variation as in [20]. Other non-local textural reconstructions could also be processed, as with the median filter and also using patch gradients or other new patch regularizations.

References

1. Arias, P., Facciolo, G., Caselles, V., Sapiro, G.: A Variational Framework for Exemplar-based Image Inpainting. IJCV **93** (2011)
2. Aubert, G., Deriche, R., Kornprobst, P.: Computing Optical Flow via Variational Techniques. SIAM J. on Appl. Math. **60** (1999)
3. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. ACM TOG **28** (2009)
4. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image Inpainting. In: Proc. SIGGRAPH (2000)
5. Buades, A., Lisani, J.L.: Video Denoising with Optical Flow Estimation. IPOL **8** (2018)
6. Bugeau, A., Bertalmio, M.: Combining Texture Synthesis and Diffusion for Image Inpainting. In: Proc. VISAPP (2009)
7. Burger, M., Dirks, H., Schönlieb, C.B.: A Variational Model for Joint Motion Estimation and Image Reconstruction. SIAM J. on Imag. Sc. **11** (2018)
8. Buyssens, P., Daisy, M., Tschumperlé, D., Lézoray, O.: Exemplar-based Inpainting: Technical Review and New Heuristics for Better Geometric Reconstructions. IEEE TIP **24** (2015)
9. Cao, F., Gousseau, Y., Masnou, S., Pérez, P.: Geometrically guided exemplar-based inpainting. SIAM J. on Appl. Math. **4** (2011)
10. Chambolle, A., Pock, T.: A First-order Primal-dual Algorithm for Convex Problems with Applications to Imaging. JMIV **40** (2011)
11. Chan, T.F., Shen, J.: Local Inpainting Models and TV Inpainting. SIAM J. on Appl. Math. **62** (2001)
12. Cocquerez, J.P., Chanas, L., Blanc-Talon, J.: Simultaneous Inpainting and Motion Estimation of Highly Degraded Video-sequences. In: Proc. SCIA (2003)
13. Criminisi, A., Pérez, P., Toyama, K.: Region Filling and Object Removal by Exemplar-Based Image Inpainting. IEEE TIP **13** (2004)
14. Do, V., Lebrun, G., Malapert, L., Smet, C., Tschumperlé, D.: Inpainting d'images couleurs par lissage anisotrope et synthèse de textures. In: Proc. RFIA (2006)
15. Efros, A.A., Leung, T.K.: Texture Synthesis by Non-parametric Sampling. In: Proc. ICCV (1999)
16. Horn, B.K., Schunck, B.G.: Determining Optical Flow. AI **17** (1981)
17. Lauze, F., Nielsen, M.: A Variational Algorithm For Motion Compensated Inpainting. In: Proc. BMVC (2004)
18. Lauze, F., Nielsen, M.: On Variational Methods for Motion Compensated Inpainting. arXiv preprint (2009)
19. Le, T., Almansa, A., Gousseau, Y., Masnou, S.: Motion-Consistent Video Inpainting. In: Proc. ICIP (2017)
20. March, R., Riey, G.: Analysis of a Variational Model for Motion Compensated Inpainting. Inverse Problems & Imaging **11** (2017)
21. Masnou, S., Morel, J.M.: Level Lines Based Disocclusion. In: Proc. ICIP (1998)
22. Newson, A., Almansa, A., Fradet, M., Gousseau, Y., Pérez, P.: Video Inpainting of Complex Scenes. SIAM J. on Imag. Sc. **7** (2014)
23. Zach, C., Pock, T., Bischof, H.: A Duality Based Approach for Realtime TV-L1 Optical Flow. In: Proc. Joint Pattern Recognition Symposium (2007)