Inpainting vidéo pour la restauration de films par reconstructions alternées de la structure et de la texture

Arthur RENAUDEAU[1] François LAUZE[2] Fabien PIERRE[3] Jean-François AUJOL[4] Jean-Denis DUROU[1]

[1] IRIT, UMR CNRS 5505, Université de Toulouse
[2] DIKU, Université de Copenhague, Danemark
[3] LORIA, UMR CNRS 7503, Université de Lorraine, INRIA projet Magrit
[4] IMB, UMR CNRS 5251, Université de Bordeaux, Bordeaux INP

arthur.renaudeau@irit.fr

Résumé

Nous proposons un nouveau modèle d'inpainting vidéo pour la restauration de films, qui combine la reconstruction de la structure par une méthode de diffusion et la reconstruction de la texture par une méthode de recopie de patchs. Les énergies proposées pour chacune de ces deux méthodes sont minimisées alternativement, afin de préserver la structure globale de l'image tout en affinant sa texture. Alors que la reconstruction de la structure est effectuée conjointement à l'estimation du mouvement par flux optique via plusieurs approches proximales, la reconstruction de la texture est traitée par une approche variationnelle non locale (NL-means). Les résultats sur différentes séquences d'images de la base de données Middlebury et de la Cinémathèque de Toulouse montrent une amélioration dans la qualité des reconstructions.

Mots Clef

Inpainting vidéo, flux optique, méthodes variationnelles.

Abstract

We propose a new video inpainting model for movies restoration application. This 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 adding textural refinement. While the structural reconstruction is obtained jointly with optical flow computation thanks to several proximal approaches, the textural reconstruction is processed by a variational non-local approach (NL-means). Results on different frames from the Middlebury database and from the Cinémathèque de Toulouse show quality improvement in the reconstructions.

Keywords

Video inpainting, optical flow, variational methods.

1 Introduction

L'*inpainting* vidéo est un problème clé de l'industrie cinématographique, qui peut aider à automatiser la restauration des films ayant subi des dégradations importantes (voir FIGURE 1), ou à mettre en œuvre des effets spéciaux nécessitant le retrait de certains éléments (« réalité diminuée »). L'*inpainting* vidéo, comme tout traitement vidéo, est de plus en plus utilisé grâce à l'augmentation des performances des processeurs et des GPU qui permet d'effectuer des calculs à grande échelle. Les techniques d'*inpainting* vidéo utilisent soit des méthodes de diffusion avec estimation du mouvement, soit des méthodes reposant sur la manipulation de patchs 3D afin de prendre en compte la redondance temporelle entre images consécutives, mais sans estimation explicite du mouvement dans ce cas. Or, le mouvement pouvant donner beaucoup d'informations pour récupérer des données, il s'agit d'un indice précieux pour effectuer l'*inpainting*. Cependant, pour estimer le mouvement, il est nécessaire de disposer de données complètes. C'est pourquoi cette estimation doit être traitée en même temps que l'*inpainting* lui-même, ce qui représente notre principal objectif.

FIGURE 1 – Exemple d'une séquences de trois images numérisées issues d'un vieux film de la Cinémathèque de Toulouse, dont l'image centrale comporte un défaut.

Dans cet article, nous visons à restaurer les défauts de type « taches » sur des films déjà numérisés. Chacun de ces défauts apparaît dans une seule image, et non sur les images situées juste avant ou juste après dans la séquence. Pour les éliminer, notre approche consiste à combiner un modèle

d'*inpainting* vidéo par diffusion calculant conjointement le flux optique, avec un modèle utilisant des patchs 2D et des cartes de correspondances vers les images voisines temporellement. Grâce à cette approche, notre modèle n'a donc besoin que des deux images adjacentes pour reconstruire la zone endommagée. Si chaque modèle pris séparément présente des inconvénients en termes de qualité de reconstruction, leur combinaison permet d'améliorer les résultats.

Après avoir décrit l'état de l'art dans la section 2, nous présentons notre modèle combiné dans la section 3. Notre stratégie numérique pour résoudre ce problème variationnel, présentée dans la section 4, consiste à optimiser alternativement les modèles utilisant la diffusion et les patchs. Les résultats de notre méthode d'*inpainting* vidéo sur différentes séquences, présentés dans la section 5, confirment l'intérêt d'une telle combinaison de modèles.

2 État de l'art

L'*inpainting* est le nom donné à la technique qui consiste à remplir les parties endommagées ou manquantes d'une image. Le terme « *inpainting* » n'a été utilisé qu'à partir de 2000 dans [4], par analogie avec le processus de restauration utilisé dans le domaine de l'art, et après celui de « désoccultation » dans [21] en 1998. Les premières applications de l'*inpainting* proviennent à l'origine de modèles de diffusion pour le débruitage, qui remontent au début des années 1990. Ce champ de recherche a été très actif ces dernières années, stimulé par de nombreuses applications : suppression de rayures ou de texte superposé à une image, restauration d'une image altérée suite à une transmission, élimination d'objets dans le contexte de la réalité diminuée.

Le remplissage de la zone à restaurer est un problème inverse mal posé, car il n'existe pas de solution unique bien définie. Il est donc nécessaire d'introduire une connaissance a priori dans le modèle. Toutes les méthodes existantes sont guidées par l'hypothèse que les pixels situés dans les parties connues et manquantes de l'image partagent les mêmes propriétés statistiques ou structures géométriques. Cette hypothèse se retrouve dans différentes hypothèses a priori, locales ou globales, afin d'obtenir une image restaurée qui soit visuellement plausible.

L'*inpainting* par diffusion vise à propager l'information contenue dans les pixels, depuis le bord de la zone endommagée vers l'intérieur de celle-ci. La variation totale pour l'*inpainting* a été introduite dans [11] pour bloquer la diffusion sur les contours des objets et induire des restaurations constantes par morceaux. L'extension de l'*inpainting* par diffusion à la vidéo a commencé dans [12] et [17, 18], avec une estimation conjointe du mouvement pour guider le remplissage de la zone endommagée. À partir du modèle de flux optique bien connu de [16] avec une régularisation lisse L^2, [2] a opté pour des normes L^1 afin de préserver les discontinuités du mouvement, en effectuant la résolution avec les algorithmes proximaux de [23]. Très récemment, [7] a choisi un modèle TV-L^1 pour résoudre l'estimation du mouvement et la reconstruction d'images, en utilisant également des algorithmes proximaux.

Cependant, ces modèles par diffusion sont limités car ils ne peuvent pas traiter les textures. C'est pourquoi des modèles par recopie complète ou partielle de patchs (détaillés dans [8]) ont été développés pour restaurer les détails, d'abord pour la synthèse de texture dans [15], puis pour l'*inpainting* local (la recherche de patchs s'effectue dans un voisinage du défaut) dans [13] avec priorité au remplissage selon l'intensité du gradient spatial sur les bords de la zone à restaurer. Enfin, les méthodes récentes recourent à un mélange de patchs suivant une recherche spatiale non locale comme dans [1]. Cette approche a également été étendue aux vidéos dans [22] et [19] avec des patchs 3D, ce qui permet une recherche de similarité temporelle dans les comparaisons entre patchs. Si ces modèles permettent de mieux récupérer les textures, ils sont cependant très dépendants de l'initialisation de la zone à remplir, afin de ne pas rester bloqué dans un minimum local suite aux différentes étapes de minimisation et entraîner une mauvaise reconstruction des structures régulières. De plus, la prise en compte de la taille du patch est également un critère important pour récupérer des textures ayant des propriétés statistiques différentes.

Dans le contexte de l'*inpainting* d'images, l'idée de mélanger les deux approches a déjà été étudiée dans [14], où la diffusion et le remplissage de la texture sont traités séquentiellement, dans [6] où l'image est décomposée en « cartoon » et « texture », qui sont restaurées séparément, et dans [9] où la reconstruction de la texture est guidée par les lignes de niveaux. Dans le contexte du débruitage vidéo, [5] utilise des patchs combinés au calcul du flux optique de [23]. Notre but dans cet article est également de tirer le meilleur parti des deux approches, en combinant la diffusion pour récupérer la structure et les patchs pour récupérer la texture, et en utilisant également les approches par diffusion et recopie de patchs pour estimer le mouvement.

3 Énoncé du problème

Définissons une séquence de trois images couleur successives $\{u_b, u, u_f\}$ comme des fonctions $\Omega \subset \mathbb{R}^2 \to \mathbb{R}^3$ à variation bornée, où u_b est l'image arrière, u est l'image courante (ou centrale) contenant le défaut à corriger et u_f l'image avant. Cette zone de défaut est définie par $O \subset \Omega$. Le problème est alors le suivant :

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

où E_S l'énergie de reconstruction de la structure, minimisée canal par canal (ou avec la luminance pour l'estimation

du mouvement), et E_T celle de reconstruction de la texture, minimisée directement sur l'image couleur. Les différentes variables apparaissant dans (1) sont explicitées dans les prochaines sous-sections.

3.1 Énergie de reconstruction de la structure

Le modèle choisi pour E_S se fonde sur les travaux de [18] et [7], en ajoutant un traitement symétrique des flux optiques avant et arrière (rappelons que u comporte ici un seul canal) :

$$E_S(v_b, u, v_f) = \mu \int_\Omega |\nabla u(x)| \, \mathrm{d}x + \lambda \int_\Omega \left| J v_b(x) \right| \mathrm{d}x + \lambda \int_\Omega \left| J v_f(x) \right| \mathrm{d}x + \int_\Omega \left| u_b(x + v_b(x)) - u(x) \right| \mathrm{d}x + \int_\Omega \left| u_f(x + v_f(x)) - u(x) \right| \mathrm{d}x \tag{2}$$

sous la contrainte $u = u^0$ sur $O^c = \Omega \backslash O$, afin de préserver la partie saine de l'image. Les termes faisant intervenir le champ de déplacements $v_b : \Omega \to \mathbb{R}^2$ (respectivement v_f) traduisent la contrainte de flux optique régularisée en norme L^1, proposée par [23], entre l'image courante u et l'image arrière u_b (respectivement l'image avant u_f), mais en utilisant des matrices jacobiennes, comme dans [10], pour conserver un caractère isotrope à la régularisation des déplacements. Chaque intégrale contient une norme discrète $|.|$ définie par $|M| = \sqrt{\sum_{i,j} m_{i,j}^2}$, qui représente une valeur absolue, une norme vectorielle ou une norme de Frobenius, selon le cas. Les paramètres λ et μ sont utilisés pour définir le compromis entre données et régularisations.

3.2 Énergie de reconstruction de la texture

La deuxième énergie E_T est une extension à la vidéo du travail de [1] utilisant directement les images couleur, et non canal par canal comme pour la reconstruction de la structure. Ici la recherche de patchs optimaux n'est plus effectuée dans un voisinage spatial autour du défaut, mais dans un voisinage temporel (dans les images précédente u_b et suivante u_f). Alors que, dans [22] et [19], la recherche de patchs 3D n'est pas limitée dans le temps, nous nous restreignons ici aux patchs 2D dans les images avant et arrière. Dans l'image courante u, le pixel central x du patch $p_u(x)$ pour lequel on recherche le patch optimal dans les images voisines u_b et u_f se situe dans une zone $\widetilde{O}$ égale à l'extension de O d'une demi-largeur de patch, afin de propager les patchs contenant suffisamment de données « saines » :

$$E_T(\Gamma_b, u, \Gamma_f) = \int_{\widetilde{O}} \omega(x) \, \varepsilon \Big[p_{u_b}(\Gamma_b(x)) - p_u(x) \Big] \, \mathrm{d}x + \int_{\widetilde{O}} (1 - \omega(x)) \, \varepsilon \Big[p_{u_f}(\Gamma_f(x)) - p_u(x) \Big] \, \mathrm{d}x \tag{3}$$

où Γ_b et Γ_f désignent les cartes de correspondances, respectivement, entre u et u_b et entre u et u_f, où $\omega(x) \in [0, 1]$ est un poids entre les deux reconstructions possibles de u à partir des images avant ou arrière (voir Section 4.4) et où la fonction ε désigne la distance entre patchs. En pratique, ε désigne une convolution entre la différence au carré des patchs $(p_{u_b}(\Gamma_b(x)) - p_u(x))^2$ et un noyau gaussien g_a d'écart-type a pour une reconstruction non locale :

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

où $x_p \in \Omega_p$ correspond aux coordonnées d'un pixel dans le patch $p_u(x)$, relativement à son centre x. Minimiser la somme de E_S et E_T constitue un problème très complexe, sans preuve d'existence ou d'unicité d'une solution, mais on cherche seulement ici à obtenir une approximation numérique de la solution en minimisant E_S et E_T de manière alternée, en utilisant le résultat de la minimisation en u de l'un des deux termes pour initialiser la minimisation de l'autre terme.

4 Optimisation

Pour restaurer par *inpainting* des défauts de grande taille ou estimer le mouvement, une approche multirésolution est nécessaire, de la résolution la plus grossière ($L = L_{\text{max}}$) vers la résolution la plus fine ($L = 0$). Dans le contexte de la vidéo, il est d'autant plus important de suivre cette stratégie pour pouvoir estimer correctement le mouvement.

4.1 Combinaison des deux modèles

Algorithme 1 - Reconstruction aux niveaux L et $L - 1$

si $L \geq L_{\text{min}}^{\text{structure}}$ **alors**
  $v_b^*, u^*, v_f^* \leftarrow \underset{v_b, u, v_f}{\text{argmin}} \left\{ E_S\left(v_b, u, v_f\right) \right\}$
  $v_b, v_f \leftarrow$ suréchantillonnage(v_b^*, v_f^*)
  $u \leftarrow u^*$
fin si
si $L \leq L_{\text{max}}^{\text{texture}}$ **alors**
  $\Gamma_b^*, \Gamma_f^* \leftarrow \underset{\Gamma_b, \Gamma_f}{\text{argmin}} \left\{ E_T\left(\Gamma_b, u, \Gamma_f\right) \right\}$
  $\Gamma_b, u, \Gamma_f \leftarrow$ suréchantillonnage$(\Gamma_b^*, u, \Gamma_f^*)$
  $u^* \leftarrow \underset{u}{\text{argmin}} \left\{ E_T\left(\Gamma_b, u, \Gamma_f\right) \right\}$
  $u \leftarrow u^*$
sinon
  $u \leftarrow$ suréchantillonnage(u)
fin si

L'idée est de suffisamment sous-échantillonner les images pour pouvoir considérer que les mouvements dans la scène sont petits. À un niveau de résolution L proche de

L_{max}, en utilisant un filtrage gaussien pour éliminer les hautes fréquences dans l'étape de sous-échantillonnage, la reconstruction de la structure fonctionne bien, alors que la reconstruction de la texture n'est pas efficace. En revanche, à plus haute résolution (L proche de 0), nous voulons mettre l'accent sur la texture. C'est pourquoi l'algorithme peut choisir un niveau maximal de résolution $L_{\text{max}}^{\text{texture}}$ pour démarrer la reconstruction de la texture et un niveau minimal de résolution $L_{\text{min}}^{\text{structure}}$ pour arrêter la reconstruction de la structure, avec $L_{\text{max}}^{\text{texture}} \geq L_{\text{min}}^{\text{structure}} - 1$ pour garantir qu'au moins une des reconstructions est effectuée à chaque résolution. Par conséquent, à chaque niveau de résolution, sauf au niveau de plus haute résolution $L = 0$, notre algorithme applique une seule reconstruction ou les deux à la fois.

Les différents suréchantillonnages sont opérés de façons différentes suivant les cas : une interpolation au plus proche voisin est appliquée aux cartes de correspondances, alors que l'interpolation est bicubique pour les autres variables. Les différentes minimisations de u, v_{b}, v_{f}, Γ_{b} et Γ_{f} de l'Algorithme 1 sont détaillées ci-après.

4.2 Estimation des mouvements

Afin de minimiser E_S par rapport au vecteur de mouvement v_{b}, on procède comme dans [23] en linéarisant l'argument des deux différences absolues dans (2). Cependant, cette linéarisation n'est possible que si les déplacements sont petits. C'est pourquoi un autre vecteur de mouvement constant v_{b}^{0} (respectivement v_{f}^{0}) est introduit, proche de v_{b}, autour duquel ce dernier est estimé :

$$\begin{aligned} E_S(v_{b}, u, v_{f}) = \mu\int_{\Omega}|\nabla u(x)|\,\mathrm{d}x + \lambda\int_{\Omega}\left|Jv_{b}(x)\right|\,\mathrm{d}x + \lambda\int_{\Omega}\left|Jv_{f}(x)\right|\,\mathrm{d}x \\ + \int_{\Omega}\Big|\nabla u_{b}(x+v_{b}^{0}(x))\cdot[v_{b}-v_{b}^{0}](x) + u_{b}(x+v_{b}^{0}(x)) - u(x)\Big|\,\mathrm{d}x \\ + \int_{\Omega}\Big|\nabla u_{f}(x+v_{f}^{0}(x))\cdot[v_{f}-v_{f}^{0}](x) + u_{f}(x+v_{f}^{0}(x)) - u(x)\Big|\,\mathrm{d}x \end{aligned} \tag{5}$$

où $\nabla u_{b}(x+v_{b}^{0}(x))\cdot[v_{b}-v_{b}^{0}](x)+u_{b}(x+v_{b}^{0}(x))-u(x)$ sera noté $\rho(u, v_{b}, u_{b}, u_{b})$ dorénavant (idem pour v_{f}). Minimiser E_S relativement au vecteur de mouvement v_{b} conduit à l'expression suivante :

$$\begin{aligned} v_{b}^{*} = \underset{v_{b}}{\operatorname{argmin}}\max_{y}\int_{\Omega}|\rho(u, v_{b}, u_{b})|\,\mathrm{d}x \\ + \langle\, Jv_{b}\,|\,y\,\rangle - \iota_{B^{\infty}}\left(\frac{y}{\lambda}\right) \end{aligned} \tag{6}$$

où nous avons introduit la variable duale de v_{b}, $y : \Omega \to \mathbb{R}^{2\times 2}$. Ce problème convexe peut être résolu par l'algorithme primal-dual de [10]. En remarquant que $Jv_{b} = [\nabla v_{b,1}, \nabla v_{b,2}]^{\top}$, l'opérateur adjoint de la jacobienne de v_{b} prend alors la forme $J^{*}y = -[\mathrm{div}([y_{1,1}, y_{1,2}]^{\top}), \mathrm{div}([y_{2,1}, y_{2,2}]^{\top})]^{\top}$. Alors que l'opérateur proximal associé à y est une projection sur la boule L^{∞}, l'opérateur proximal associé à v_{b} est un seuillage doux (voir [23] pour plus de détails) :

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

où $\sigma, \tau > 0$ sont des pas de temps et $\theta \in [0, 1]$. La minimisation de v_{f} s'effectue de la même manière.

4.3 Reconstruction de la structure

Une fois le mouvement estimé, le processus d'*inpainting* est obtenu grâce à la minimisation suivante :

$$\begin{aligned} u^{*} = \underset{u}{\operatorname{argmin}}\int_{\Omega}|u_{b}(x+v_{b}(x)) - u(x)|\,\mathrm{d}x \\ + \int_{\Omega}\left|u_{f}(x+v_{f}(x)) - u(x)\right|\,\mathrm{d}x + \mu\int_{\Omega}|\nabla u(x)|\,\mathrm{d}x \end{aligned} \tag{8}$$

Pour réécrire le problème convexe (8) avec des variables duales, la dépendance temporelle de u, v_{b} et v_{f} doit être clarifiée. En effet, en prenant 1 comme pas de temps entre deux images, et en introduisant la variable de temps t dans u, les fonctions u_{b} et u_{f} prennent la forme suivante :

$$\begin{aligned} u_{b}(x+v_{b}(x)) &= u(x+v_{b}(x,t), t-1) = u\left(\varphi_{b}(x,t)\right) \\ u_{f}(x+v_{f}(x)) &= u(x+v_{f}(x,t), t+1) = u\left(\varphi_{f}(x,t)\right) \end{aligned} \tag{9}$$

où φ_{b} et φ_{f} sont deux transformations supposées différentiables et inversibles, similaires aux cartes de correspondances Γ_{b} et Γ_{f}, en faisant l'hypothèse que $\varphi_{b}\circ\varphi_{f} = \varphi_{f}\circ\varphi_{b} = I_{d}$ presque partout. Avec ces nouvelles notations, (8) peut être réécrite comme :

$$\begin{aligned} u^{*} = \underset{u}{\operatorname{argmin}}\max_{z}\left\langle \begin{array}{c} u\circ\varphi_{b} - u \\ u\circ\varphi_{f} - u \\ \nabla u \end{array} \middle| z \right\rangle \\ -\iota_{B^{\infty}}(z_{1}) - \iota_{B^{\infty}}(z_{2}) - \iota_{B^{\infty}}\left(\frac{1}{\mu}[z_{3}, z_{4}]^{\top}\right) \end{aligned} \tag{10}$$

où nous introduisons la variable duale de u, $z : \Omega \to \mathbb{R}^{4}$. En notant K l'opérateur relatif à u dans le produit scalaire, cela conduit à l'opérateur adjoint K^{*} décrit dans [18] :

$$\begin{aligned} K^{*}z = \det\left(J\varphi_{f}\right) z_{1}\circ\varphi_{f} - z_{1} + \det\left(J\varphi_{b}\right) z_{2}\circ\varphi_{b} \\ -z_{2} - \mathrm{div}([z_{3}, z_{4}]^{\top}) \end{aligned} \tag{11}$$

La minimisation (10) peut également être effectuée en utilisant l'algorithme primal-dual de [10] :

$$\begin{cases} z_{1}^{(n+1)} \leftarrow \mathrm{prox}_{\sigma\iota_{B^{\infty}}}\left(z_{1}^{(n)} + \sigma\left(u\circ\varphi_{b} - \bar{u}^{(n)}\right)\right) \\ z_{2}^{(n+1)} \leftarrow \mathrm{prox}_{\sigma\iota_{B^{\infty}}}\left(z_{2}^{(n)} + \sigma\left(u\circ\varphi_{f} - \bar{u}^{(n)}\right)\right) \\ \begin{bmatrix} z_{3}^{(n+1)} \\ z_{4}^{(n+1)} \end{bmatrix} \leftarrow \mathrm{prox}_{\mu\,\sigma'\iota_{B^{\infty}}}\left(\begin{bmatrix} z_{3}^{(n)} \\ z_{4}^{(n)} \end{bmatrix} + \sigma'\nabla\bar{u}^{(n)}\right) \\ u^{(n+1)} \leftarrow u^{(n)} - \tau K^{*}z^{(n+1)} \\ \bar{u}^{(n+1)} \leftarrow u^{(n+1)} + \theta\left(u^{(n+1)} - u^{(n)}\right) \end{cases} \tag{12}$$

où $\sigma, \sigma', \tau > 0$ sont des pas de temps et $\theta \in [0,1]$. Pour minimiser une énergie similaire à (2), [7] répète les minimisations alternées pour l'*inpainting* de u et l'estimation d'un unique champ de déplacements v jusqu'à la convergence. A contrario, dans notre cas, les trois variables $(v_{_b}, v_{_f}, u)$ sont minimisées simultanément, en appliquant des étapes proximales successives sur chaque variable. Les trois descentes suivant $v_{_b}$, $v_{_f}$ et u donnent de meilleures résultats que la première option, pour un temps de calcul équivalent.

4.4 Reconstruction de la texture avec estimation des cartes de correspondances

La minimisation de l'énergie de texture E_T se fonde sur les travaux de [1], où les cartes de correspondances $\Gamma_{_b}$ et $\Gamma_{_f}$ sont estimées en utilisant l'algorithme PatchMatch de [3] pour opérer une recherche non locale efficace sans être exhaustive, avec une distance L^2 entre patchs. Par conséquent, pour $\Gamma_{_b}$ (respectivement $\Gamma_{_f}$), on a, $\forall x \in \widetilde{O}$:

$$\Gamma_{_b}(x) = \underset{x_b \in \Omega}{\operatorname{argmin}} \int_{\Omega_p} g_a(x_p)\ [u_{_b}(x_b - x_p) - u(x - x_p)]^2 \,\mathrm{d}x_p \tag{13}$$

Pour chacune des images voisines $u_{_b}$ et $u_{_f}$, chacun des termes du membre de droite de (3) peut être réécrit, sans prendre en compte le poids ω dans un premier temps, comme le cas extrême où l'on choisit seulement le patch le plus proche pour chaque pixel de la zone de défaut (voir [1] pour plus de détails), en introduisant une fonction de Dirac δ :

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

Avec les changements de variables $x := x - x_p$ et $x_b := x_b - x_p$, qui opèrent deux translations, on tire de (4) et (14) :

$$E_T^b(u, \Gamma_{_b}) = \int_{O} \int_{\Omega} m(x, x_b)\, [u_{_b}(x_b) - u(x)]^2 \,\mathrm{d}x_b \,\mathrm{d}x \tag{15}$$

où :

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

dont l'intégrale sur Ω est égale à 1, puisque g_a est, par hypothèse, une gaussienne normalisée. En développant la différence au carré dans (15), u est aussi le minimiseur de l'énergie suivante, qui est égale à $E_T^b(u, \Gamma_{_b})$ à une constante près :

$$\widetilde{E}_T^b(u, \Gamma_{_b}) = \int_{O} \left[u(x) - \int_{\Omega} m(x, x_b)\, u_{_b}(x_b) \,\mathrm{d}x_b \right]^2 \mathrm{d}x \tag{17}$$

ce qui nous fournit directement la solution des moyennes non locales définie, $\forall x \in O$, par :

$$\begin{aligned} u(x) &= \int_{\Omega} m(x, x_b)\, u_{_b}(x_b) \,\mathrm{d}x_b \\ &= \int_{\Omega_p} g_a(x_p)\, u_{_b}(\Gamma_{_b}(x + x_p) - x_p) \,\mathrm{d}x_p \end{aligned} \tag{18}$$

Il s'agit ici du résultat pour une seule des deux images voisines. Pour prendre en compte les deux images, des moyennes pondérées peuvent être effectuées avec les poids $\omega(x)$ et $(1 - \omega(x))$. Moyenner les deux résultats ($\omega(x) = 0.5$, notée T_H) provoquera l'apparition de flou. À l'inverse, choisir le meilleur des deux résultats ($\omega(x) \in \{0, 1\}$, notée T_B) fera apparaître des artéfacts spatiaux. Dans ce cas, le choix entre 0 et 1 pour les poids est effectué après avoir estimé les deux cartes de correspondances $\Gamma_{_b}$ et $\Gamma_{_f}$. Ensuite, pour chaque pixel à reconstruire, le poids $\omega(x)$ est égal à 1 si la distance estimée $\varepsilon \left[p_{u_{_b}}(\Gamma_{_b}(x)) - p_u(x) \right]$ au meilleur patch dans l'image arrière est inférieure à la distance estimée $\varepsilon \left[p_{u_{_f}}(\Gamma_{_f}(x)) - p_u(x) \right]$ au meilleur patch dans l'image avant, et est égal à 0 sinon. Une solution intermédiaire consiste à appliquer une moyenne pondérée (notée T_W) en utilisant le ratio entre la distance estimée au meilleur patch dans l'image avant et la somme des deux distances estimées.

5 Expérimentations

Notre algorithme a été implémenté en Matlab et en C. Le niveau maximal de la pyramide multirésolution vaut $L_{\text{max}} = 10$, et le facteur entre étages successifs est égal à $\sqrt{2}$. L'algorithme est d'abord testé sur trois images consécutives issues de séquences Middlebury, où un défaut a été introduit artificiellement dans l'image centrale (voir FIGURE 2). Il est ensuite testé en conditions réelles, sur des images numérisées issues de vieux films de la Cinémathèque de Toulouse.

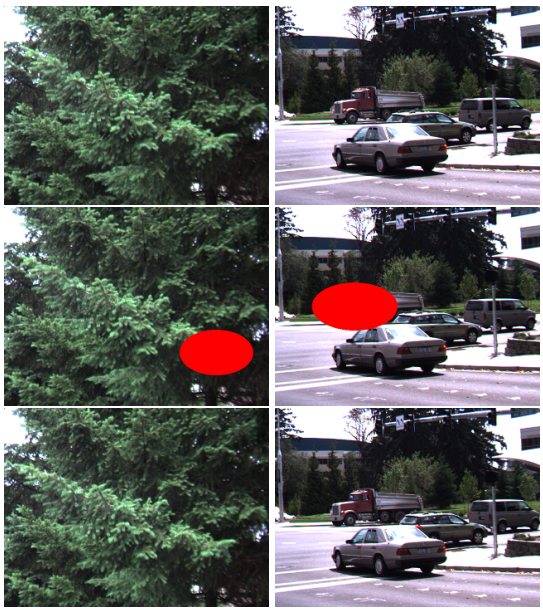
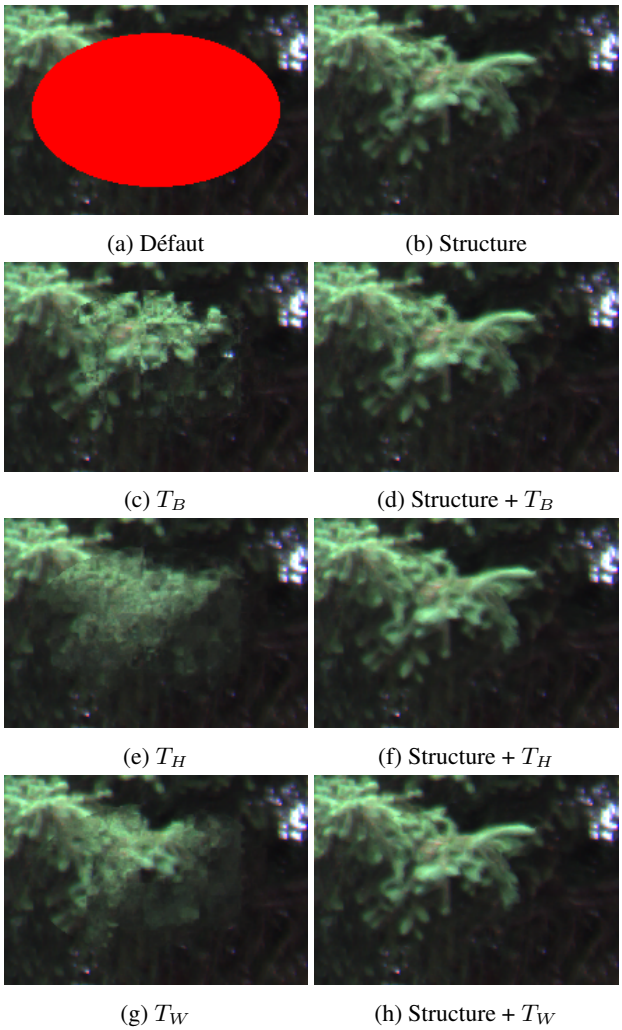


FIGURE 2 – Images 8 à 10 des séquences « Evergeen » (gauche) et « Dumptruck » (droite) de la base Middlebury. Le défaut est surligné en rouge dans l'image centrale.

Les vidéos des séquences complètes de Middlebury contenant 8 images et comportant le défaut artificiel, ainsi que les reconstructions correspondantes, sont disponibles en téléchargement sur le lien suivant : *ORASIS-VideoInpainting1*.

5.1 Séquence « Evergreen »

Pour ce premier test, les résultats de la FIGURE 3 montrent que les méthodes où l'on ne reconstruit que la texture (3c, 3e, 3g) ne parviennent pas à restituer correctement les branches de l'arbre. L'approche avec le meilleur des deux patchs (3c) provoque des artéfacts, alors que les approches utilisant un moyennage (3e et 3g) conduisent à un résultat où la texture est perdue au profit d'une homogénéité non désirée. La reconstruction de la structure (3b) est meilleure, mais encore un peu floue à cause de la diffusion, et certaines branches semblent un peu trop étirées. Enfin, la combinaison des deux reconstructions (3d, 3f, 3h) réussit à superposer une texture à la structure.



(a) Défaut (b) Structure

(c) T_B (d) Structure + T_B

(e) T_H (f) Structure + T_H

(g) T_W (h) Structure + T_W

FIGURE 3 – Image centrale « Evergreen » : zooms sur les reconstructions de la zone défectueuse.

Ces premières interprétations visuelles sont confirmées par les mesures de qualité du TABLEAU 1. La structure est plus performante que la texture, tandis que notre algorithme donne les meilleurs résultats, avec un écart non négligeable. Par ailleurs, le choix de l'une quelconque des trois reconstructions de la texture, associée à la reconstruction de la structure, n'a pas de réel impact sur le résultat.

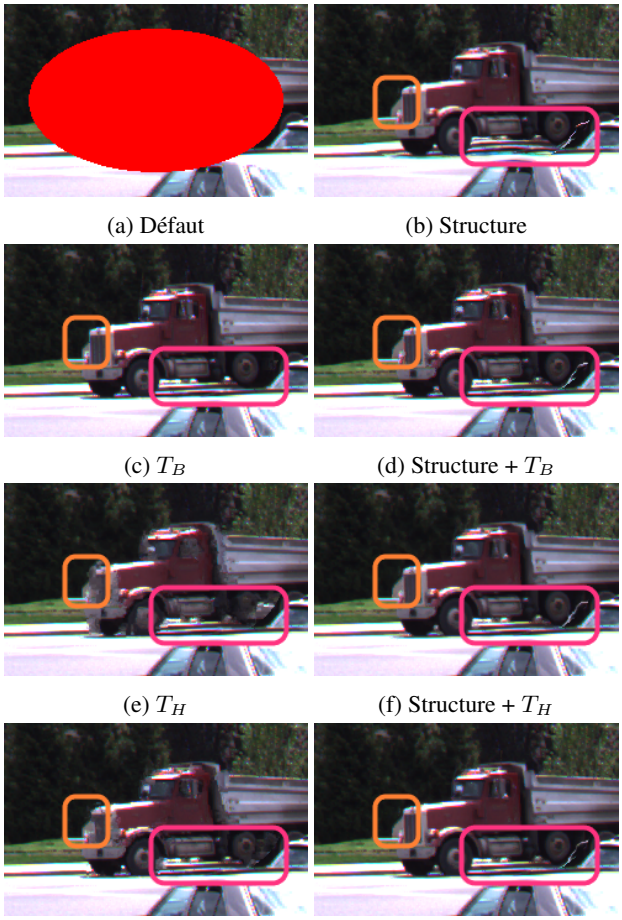
	Sans structure	Structure
Sans texture	18.66 - 0.943	37.24 - 0.991
Texture T_B	30.94 - 0.978	**40.37** - **0.994**
Texture T_H	33.37 - 0.980	**40.47** - **0.994**
Texture T_W	33.56 - 0.982	**40.50** - **0.994**

TABLEAU 1 – Mesures de PSNR - SSIM pour les différentes reconstructions de l'image centrale « Evergreen ».

5.2 Séquence « Dumptruck »

Pour ce deuxième test, les résultats de la FIGURE 4 montrent que la reconstruction de la structure (4b) entraîne une déformation du camion, qui semble osciller verticalement dans la vidéo. De plus, il reste une trace derrière la voiture se dirigeant vers la droite. Les reconstructions de la texture à l'aide de moyennes (4e et 4g) conduisent à une reconstruction floue du camion. Même en utilisant le meilleur patch des deux images adjacentes (4c), dont le résultat semble être bon sur l'image reconstruite seule, la vidéo montre que l'algorithme choisit l'image la plus proche, en terme de distance entre patchs, et reste « verrouillé ». C'est pourquoi il semble y avoir un manque de mouvement dans la vidéo, à l'intérieur de la zone de défaut. Notre algorithme (4d, 4f, 4h) fournit de meilleurs résultats (il reste quand même une trace à l'arrière de la voiture), grâce à l'apport de la texture qui permet de conserver un bon flux optique.

D'autre part, les mesures de qualité du TABLEAU 2 montrent que notre algorithme améliore significativement les résultats, en comparaison des reconstructions obtenues avec la structure seule ou avec la texture seule.



(a) Défaut (b) Structure

(c) T_B (d) Structure + T_B

(e) T_H (f) Structure + T_H

(g) T_W (h) Structure + T_W

FIGURE 4 – Image centrale « Dumptruck » : zooms sur les reconstructions de la zone défectueuse. Dans la zone encadrée en magenta, la structure seule (4b) déforme le dessous du camion. La reconstruction près de la voiture de droite reste problématique dans tous les cas. Dans la zone encadrée en orange, la texture avec choix du meilleur patch (4c) décale le camion vers la gauche.

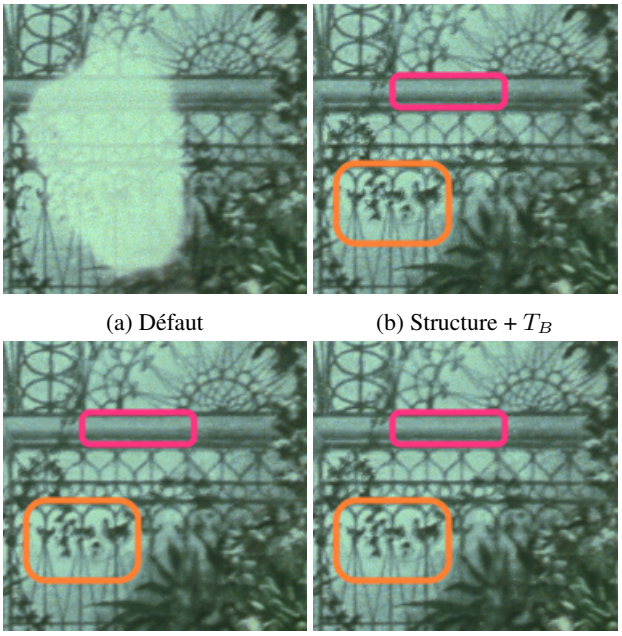
	Sans structure	Structure
Sans texture	17.53 - 0.937	27.76 - 0.975
Texture T_B	33.16 - 0.987	**38.11** - **0.994**
Texture T_H	34.35 - 0.989	**38.87** - **0.994**
Texture T_W	34.31 - 0.988	**38.58** - **0.994**

TABLEAU 2 – Mesures de PSNR - SSIM pour les différentes reconstructions de l'image centrale « Dumptruck ».

5.3 Séquence de la Cinémathèque

FIGURE 5 – Sélection manuelle d'un masque englobant les zones défectueuses : image comportant des traces blanches (gauche), et sélection en rouge du masque englobant les défauts (droite).

Les tests sur des séquences comportant un défaut artificiel ont montré que la combinaison de la structure et de la texture améliorait effectivement les reconstructions. L'enjeu de l'algorithme proposé est de pouvoir corriger les défauts de vidéos ayant subi une réelle dégradation. Nous utilisons pour cela les images numérisées de vieux films argentiques provenant du fonds de la Cinémathèque de Toulouse (voir FIGURE 1). La première étape consiste à déterminer manuellement les zones défectueuses sous la forme d'un masque binaire (voir FIGURE 5).



(a) Défaut (b) Structure + T_B

(c) Structure + T_H (d) Structure + T_W

FIGURE 6 – Image de la FIGURE 5 : zooms sur trois reconstructions de l'une des zones détériorées. Dans la zone encadrée en magenta, l'utilisation de la moyenne des deux patchs (6c) provoque un lissage excessif. A contrario, dans la zone encadrée en orange, le choix du meilleur des deux patchs (6b) a pour effet de trop accentuer les contours.

Une fois le défaut détecté, les trois variantes de notre algorithme, appliquées à la séquence contenant l'image de la FIGURE 5, fournissent trois reconstructions de qualité globalement satisfaisante (voir FIGURE 6). Malgré tout, la question d'un surplus de flou (6c) ou de la présence d'artéfacts (6b) se pose encore, en particulier vis-à-vis des professionnels du cinéma, pour qui la préservation du grain argentique semble cruciale. La solution intermédiaire (6d) peut donc constituer un bon compromis, car elle lisse légèrement les formes tout en évitant les artéfacts et en préservant suffisamment le grain. Sur les vidéos, la différence entre ces trois reconstructions est difficile à percevoir avec un défaut présent sur une seule image (laps de temps trop court). Les différentes vidéos sont disponibles en téléchargement sur le lien suivant : *ORASIS-VideoInpainting2*.

6 Conclusion et perspectives

Dans cet article, nous avons montré que la combinaison d'une reconstruction de la structure par des approches de diffusion et d'une reconstruction de la texture permettait d'améliorer les résultats de l'*inpainting* vidéo, en termes de qualité visuelle et de métriques. Pour aller plus loin, il serait intéressant d'affiner le modèle de flux optique en utilisant la variation totale généralisée, comme dans [20]. D'autres reconstructions non locales de la texture peuvent également être envisagées, par exemple avec le filtre médian, ou en utilisant des gradients de patchs ou de nouvelles régularisations de patchs. D'autre part, nous envisageons de recourir au *deep learning*, soit en amont pour la détection des défauts, soit pour le calcul des opérateurs proximaux de la reconstruction elle-même.

Références

[1] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A Variational Framework for Exemplar-based Image Inpainting. *IJCV*, 93, 2011.

[2] G. Aubert, R. Deriche, and P. Kornprobst. Computing Optical Flow via Variational Techniques. *SIAM Journal on Applied Mathematics*, 60, 1999.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A Randomized Correspondence Algorithm for Structural Image Editing. *ACM Transactions on Graphics*, 28, 2009.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image Inpainting. In *Proceedings of SIGGRAPH*, 2000.

[5] A. Buades and J.-L. Lisani. Video Denoising with Optical Flow Estimation. *IPOL*, 8, 2018.

[6] A. Bugeau and M. Bertalmio. Combining Texture Synthesis and Diffusion for Image Inpainting. In *Proceedings of VISAPP*, 2009.

[7] M. Burger, H. Dirks, and C.-B. Schönlieb. A Variational Model for Joint Motion Estimation and Image Reconstruction. *SIAM Journal on Imaging Sciences*, 11, 2018.

[8] P. Buyssens, M. Daisy, D. Tschumperlé, and O. Lézoray. Exemplar-based Inpainting: Technical Review and New Heuristics for Better Geometric Reconstructions. *IEEE Transactions on Image Processing*, 24, 2015.

[9] F. Cao, Y. Gousseau, S. Masnou, and P. Pérez. Geometrically guided exemplar-based inpainting. *SIAM Journal on Applied Mathematics*, 4, 2011.

[10] A. Chambolle and T. Pock. A First-order Primal-dual Algorithm for Convex Problems with Applications to Imaging. *JMIV*, 40, 2011.

[11] T. F. Chan and J. Shen. Local Inpainting Models and TV Inpainting. *SIAM Journal on Applied Mathematics*, 62, 2001.

[12] J.-P. Cocquerez, L. Chanas, and J. Blanc-Talon. Simultaneous Inpainting and Motion Estimation of Highly Degraded Video-sequences. In *Proceedings of SCIA*, 2003.

[13] A. Criminisi, P. Pérez, and K. Toyama. Region Filling and Object Removal by Exemplar-Based Image Inpainting. *IEEE Transactions on Image Processing*, 13, 2004.

[14] V. Do, G. Lebrun, L. Malapert, C. Smet, and D. Tschumperlé. Inpainting d'images couleurs par lissage anisotrope et synthèse de textures. In *Proceedings of RFIA*, 2006.

[15] A. A. Efros and T. K. Leung. Texture Synthesis by Non-parametric Sampling. In *Proceedings of ICCV*, 1999.

[16] B. K. Horn and B. G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17, 1981.

[17] F. Lauze and M. Nielsen. A Variational Algorithm For Motion Compensated Inpainting. In *Proceedings of BMVC*, 2004.

[18] F. Lauze and M. Nielsen. On Variational Methods for Motion Compensated Inpainting. *arXiv preprint*, 2009.

[19] T. Le, A. Almansa, Y. Gousseau, and S. Masnou. Motion-Consistent Video Inpainting. In *Proceedings of ICIP*, 2017.

[20] R. March and G. Riey. Analysis of a Variational Model for Motion Compensated Inpainting. *Inverse Problems & Imaging*, 11, 2017.

[21] S. Masnou and J.-M. Morel. Level Lines Based Disocclusion. In *Proceedings of ICIP*, 1998.

[22] A. Newson, A. Almansa, M. Fradet, Y. Gousseau, and P. Pérez. Video Inpainting of Complex Scenes. *SIAM Journal on Imaging Sciences*, 7, 2014.

[23] C. Zach, T. Pock, and H. Bischof. A Duality Based Approach for Realtime TV-L1 Optical Flow. In *Proceedings of Joint Pattern Recognition Symposium*, 2007.