



HAL
open science

Fondu spatial de patches pour la réduction des artéfacts dans des méthodes génériques d'inpainting "basées motif"

Maxime Daisy, David Tschumperlé, Olivier Lezoray

► To cite this version:

Maxime Daisy, David Tschumperlé, Olivier Lezoray. Fondu spatial de patches pour la réduction des artéfacts dans des méthodes génériques d'inpainting "basées motif". Colloque sur le Traitement du Signal et des Images (GRETSI'13), Sep 2013, Brest, France. pp.ID199. hal-00927558

HAL Id: hal-00927558

<https://hal.science/hal-00927558>

Submitted on 13 Jan 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Fondu spatial de patches pour la réduction des artéfacts dans des méthodes génériques d’inpainting “basées motif”

Maxime DAISY, David TSCHUMPERLÉ, Olivier LÉZORAY

GREYC Image - ENSICAEN

6, Boulevard Maréchal Juin - F-14050 CAEN cedex - FRANCE

{Maxime.Daisy, David.Tschumperle, Olivier.Lezoray}@ensicaen.fr

Résumé – L’inpainting par patch (ou basé “motif”), est une technique de traitement permettant la reconstruction de pixels manquants ou invalides dans des images. Elle s’avère particulièrement efficace pour reconstituer de larges portions d’images, notamment grâce à ses bonnes propriétés de reconstruction des zones texturées. Néanmoins, il existe toujours des configurations “pathologiques” de reconstruction qui font apparaître des artéfacts visibles sur les images inpaintées, et ce, pour tous les algorithmes basés sur de la copie de patches (moyennés ou non). Dans cet article, nous nous focalisons sur ces cas problématiques et proposons une technique générique de réduction de ces artéfacts par mélange spatial de patches, qui peut s’adapter à tous types d’algorithmes d’inpainting basés “motif”.

Abstract – Patch-based (or “pattern-based”) inpainting, is a processing technique able to reconstruct missing or invalid pixels in images. It is particularly effective to process wide image areas, thanks to its ability to reconstruct textured data. Nevertheless, there are always “pathological” reconstruction configurations that generate some visible artefacts on inpainted images, and this for all kind of pattern-based inpainting algorithms. In this paper we focus on these problematic cases and propose a generic spatial-blending technique that can be adapted to any type of patch-based inpainting methods in order to reduce these artefacts.

1 Contexte et problématique

L’inpainting est le domaine du traitement d’image dans lequel on tente de reconstituer des portions manquantes d’une image tout en assurant une cohérence géométrique maximale dans la reconstruction. Une bonne revue des méthodes d’inpainting peut être trouvée dans [3], où deux principaux types de méthodes de reconstruction sont analysées et mises en avant :

- Les méthodes orientées “géométrie” [2, 9, 12, 13], initiées par Masnou et Morel [9] pour la reconstruction de lignes de niveau, qui résolvent l’inpainting par l’application d’équations aux dérivées partielles [13], ou par la résolution d’un problème de minimisation de fonctionnelles [2, 9]. Ces méthodes permettent de prolonger localement la géométrie des structures aux frontières du domaine d’inpainting. Cependant, elles ne permettent pas de régénérer des zones texturées, ce qui fait souvent apparaître une inhomogénéité spatiale entre les données connues et les données reconstruites, notamment lorsque l’on traite des images naturelles.

- Les méthodes orientées “motif” [1, 3, 4, 5, 6, 8, 11, 15], initiées par Efros et Leung [5] avec leurs travaux sur l’extrapolation de texture, qui sont basées sur le copier-coller de patches présents dans la zone connue de l’image vers la zone inconnue à reconstruire [4]. Ces algorithmes génèrent un patchwork de morceaux d’images, en les choisissant itérativement comme les plus similaires à ceux présents à la frontière du domaine d’inpainting. Ces méthodes donnent des résultats très intéressants en terme de reconstruction de larges zones texturées.

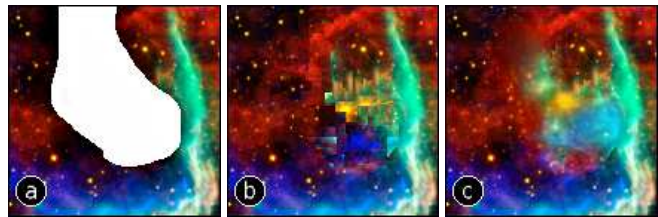


FIGURE 1 – Intérêt de notre contribution : a) image masquée à reconstruire (zoom), b) image inpaintée par copier-coller de patches (zoom), c) image obtenue après fondu de patches (zoom).

Des variantes proposent de mélanger ces deux approches pour mettre en oeuvre des méthodes hybrides [1, 7, 12], d’autres [8, 14] de moyenner plusieurs patches candidats avant de les recoller. Dans tous les cas, il existe toujours des configurations locales qui ne permettent pas de sélectionner des patches satisfaisants au critère de similarité choisi. Copier-coller ou moyenner de tels patches provoque des artéfacts visuels pouvant se manifester par des effets de bloc, ou des changements brusques d’intensité ou de textures dans l’image. Dans cet article, nous introduisons une technique originale de fondu spatial de patch pour réduire ce type d’artéfacts (cf. Fig. 1). Notre algorithme a l’intérêt de s’adapter à tous type de méthodes d’inpainting basés “motif”. Puis, nous proposons une façon de détecter automatiquement les points où les artéfacts d’inpainting sont les plus visibles. L’application simultanée de nos deux contributions est illustrée par des résultats d’inpainting par patch dont les artéfacts visuels sont réduits par rapport aux algorithmes de référence auxquels nous nous comparons.

Cet article est organisé de la manière suivante. La première partie constitue le coeur de notre méthode, à savoir la modification d’un algorithme d’*inpainting* par patch afin qu’il fournisse les données nécessaires pour notre fondu spatial de patches. La deuxième partie est consacrée à un problème de détection automatique d’artéfacts produits par un algorithme d’*inpainting* par patch. La troisième et dernière partie fournit des résultats commentés et la conclusion de cet article.

2 Fondu entre patches pour l’*inpainting*

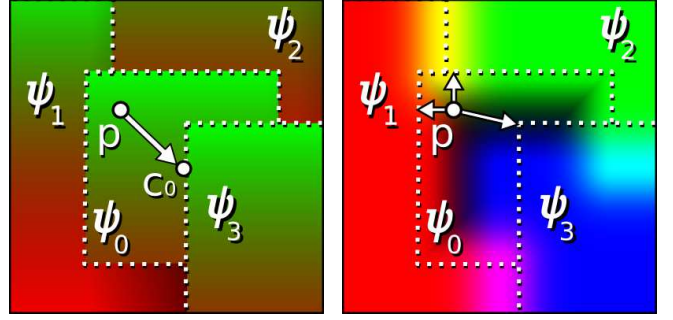
Dans cette partie, nous entrons dans le détail de la méthode permettant de créer un fondu spatial entre patches.

Soit $I : \Gamma \mapsto \mathbb{R}_+^3$ une image couleur dont on veut reconstruire des pixels supposés “invalides” définis sur un domaine $\Omega \subset \Gamma$ (masque d’*inpainting*). Nous dénotons par ψ_p un patch de l’image de taille carrée ($N \times N$ avec N impair) et centré en $p \in \Gamma$. La composante i d’un patch ψ_p sera notée $\psi_p^i \in \mathbb{R}^{N \times N}$. La plupart des algorithmes d’*inpainting* basés “motif” se fondent sur la recherche, pour chaque point $p \in \Omega$ à reconstruire, du “meilleur” patch ψ_q (avec $\psi_q \subset \Gamma/\Omega$), et sur la recopie de la partie de ψ_q qui recouvre $\psi_p \cap \Omega$ dans l’image I à reconstruire. Le contenu de Ω est ainsi itérativement reconstruit de façon globalement concentrique (certains points d’intérêt peuvent être néanmoins traités dans un ordre dépendant d’un critère de priorité choisi) [4, 8].

Notre proposition est de modifier un tel algorithme d’*inpainting* afin de pouvoir appliquer une technique de fondu de patches pour améliorer visuellement l’image I . Cette modification requiert de mémoriser à la fois la carte $\mathcal{U} : \Gamma \mapsto \Gamma$ des positions des centres des patches d’origine copiés, et la carte $\mathcal{S} : \Omega \mapsto \mathbb{R}^2$ des décalages entre les points reconstruits et les centres des patches partiellement recollés. La labélisation en composantes connexes \mathcal{L} de \mathcal{U} définit la partition des morceaux de patches connus qui ont été recollés dans Ω . À partir de ces données I , \mathcal{U} , \mathcal{S} et \mathcal{L} , nous sommes capables de générer une image $J : \Gamma \mapsto \mathbb{R}_+^3$ dont les patches de I recollés par le processus d’*inpainting* se “fondent” spatialement les uns dans les autres. En pratique, pour chaque point $p \in \Omega$, ce fondu se réalise grâce à l’ensemble Ψ_p des patches ψ_p d’un voisinage $\mathcal{V}(p)$ de p , patches pouvant potentiellement contenir p . La construction de Ψ_p se réalise par la recherche exhaustive de tous les centres de patches différents qui ont été recollés dans $\mathcal{V}(p)$. Le fondu se calcule alors en p en mélangeant spatialement tous les patches de l’ensemble Ψ_p de la façon suivante pour chaque composante i :

$$J^i(p) = \frac{\sum_{\psi_q \in \Psi_p} w(q, p) \psi_q^i(p - q)}{\varepsilon + \sum_{\psi_q \in \Psi_p} w(q, p)} \quad (1)$$

avec $\varepsilon \in \mathbb{R}$ proche de 0 stabilisant l’équation. La quantité $\psi_q^i(p - q)$ représente la valeur du pixel aux coordonnées $((N/2, N/2) + (p - q))$ dans ψ_q . Les poids gaussiens $w(q, p) = e^{-\frac{d(q, p)^2}{\sigma^2}}$ donnent plus d’importance aux patches de Ψ_p les plus proches spatialement du point p dans le calcul du mélange. La



(a) Représentation de la carte \mathcal{S} des décalages avec les centres des patches. (b) Représentation des poids w pour le calcul de (1) des points de ψ_0 .

FIGURE 2 – Illustration des données \mathcal{S} et w calculés pour le mélange de patches.

variance σ est un paramètre utilisateur relatif à l’amplitude du mélange spatial (largeur de bande). La fonction $d(p, q)$ définit la distance spatiale minimale entre le point p et la partie du patch recollée contenant q :

$$d(q, p) = \min_{q' \in \mathcal{V}(p)} \|q' - p\| \quad \text{où } \mathcal{L}(q') = \mathcal{L}(q) \quad (2)$$

L’utilisation de cette distance $d(p, q)$ permet de générer un fondu spatial orienté orthogonalement aux différentes frontières entre les patches voisins de p .

La Fig. 2 illustre une situation synthétique sur laquelle on veut appliquer cette technique de fondu de patches. p est le point dont on veut calculer la valeur, et $\psi_0 \dots \psi_3$ sont les différents patches qui composent Ψ_p . Les composantes x et y des vecteurs de \mathcal{S} sont représentés dans les canaux rouge et vert. Le point c_0 est le centre du patch recollé ψ_0 , et donc $\mathcal{S}(p) = c_0 - p$. Dans la Fig. 2(b), les flèches représentent les distances minimales de p à chacun des patches voisins. Les poids w relatifs à ψ_1, ψ_2, ψ_3 sont représentés respectivement par des intensités lumineuses dans les canaux rouge, vert et bleu ($\forall q \in \psi_0, w(p, q) = 0$).

Notre technique de mélange spatial de patches, dont l’Algorithme 1 récapitule les principales étapes, utilise tous les patches avec lesquels un point $p \in \Omega$ aurait pu être reconstruit pendant l’*inpainting*, si l’ordre de recopie des patches avait été différent de celui réellement utilisé. Intuitivement, cela revient à mémoriser une liste de patches candidats (plutôt qu’un seul) en chaque point $p \in \Omega$ reconstruit par l’algorithme d’*inpainting*. Ceci est très différent d’un processus d’*inpainting* qui utiliserait un moyennage des K meilleurs patches candidats (tel que proposé dans [8]) pour reconstruire chaque point p , puisqu’ici le mélange calculé exploite à la fois la géométrie du morceau de patch recollé en p , mais aussi la configuration spatiale de tous les patches voisins de ψ_p .

Notons que cette technique de fondu s’applique avec une largeur de bande σ constante. En réalité, toutes les zones de l’image *inpaintée* ne présentent pas des ruptures de contraste équivalentes et il est naturellement souhaitable de pouvoir ajuster localement σ , d’où l’objet de notre deuxième contribution.

Algorithme 1 : Mélange spatial de patches pour l’inpainting

```
1 pour  $p \in \Omega$  faire
2   var centres = liste( $\emptyset$ );
3   var distances = liste( $\emptyset$ );
4   pour  $q \in \mathcal{V}(p)$  faire
5     var  $c = q + \mathcal{S}(q)$ ;
6     si  $(c \notin \text{centres}) \wedge (p \in \psi_c)$  alors
7       insérer(centres,  $c$ );
8       insérer(distances,  $d(c, p)$ ); // cf Eq. (2)
9   J(p) = fondre(p, centres, distances); // cf Eq. (1)
```

3 Détection des artéfacts d’inpainting

Cette deuxième partie montre une manière empirique de détecter des artéfacts produits par des méthodes d’inpainting par patch. Nous nous basons sur une donnée supplémentaire fournie par l’inpainting par patch, à savoir \mathcal{U} , grâce à la modification d’algorithme que nous avons proposé.

La majeure partie des artéfacts visuels de reconstruction proviennent principalement du fait que des patches peu similaires entre eux ont été copiés-collés à des positions spatialement proches dans Ω . En pratique, les points $p \in \Omega$ où se produisent ces artéfacts se situent dans des zones vérifiant à la fois les deux conditions suivantes :

1. $\|\nabla I(p)\|$ est élevé, ce qui correspond à des discontinuités spatiales importantes dans les valeurs de l’image.

2. Les patches collés dans le voisinage de p proviennent de localisations différentes et éloignées, ce qui se traduit localement par de fortes discontinuités de \mathcal{U} .

Par conséquent, nous proposons de détecter l’ensemble \mathcal{E} de ces points d’artéfacts (aussi appelés *points de rupture*) comme les points vérifiant la condition $\mathcal{R}(p) > \tau$, où :

$$\forall p \in \Omega, \quad \mathcal{R}(p) = \frac{\|\nabla I(p)\| \cdot |\text{div}(\mathcal{U}(p))|}{\alpha}$$

et $\alpha = (\max_{q \in I} \|\nabla I(q)\|) \cdot (\max_{q \in I} |\text{div}(\mathcal{U}(q))|)$ est un facteur de normalisation, pour que le paramètre utilisateur τ , relatif à la densité de points d’artéfacts détectés, puisse être choisi dans $[0, 1]$. $\mathcal{R}(p)$ permet d’estimer localement la force de la rupture géométrique due à la reconstruction, puisque $|\text{div}(\mathcal{U})|$ détermine si deux patches qui ont été collés côte à côte proviennent de localisations proches ($|\text{div}(\mathcal{U})|$ faible) ou éloignées ($|\text{div}(\mathcal{U})|$ élevée). Par conséquent, si $\mathcal{R}(p)$ est élevé, le point p vérifie simultanément 1. et 2. et a de fortes chances de localiser un artéfact dans l’image I .

En associant à chaque point de rupture $p \in \mathcal{E}$ une fonction gaussienne dont la variance dépend de $\mathcal{R}(p)$, on peut définir une carte spatiale $\sigma : \Gamma \mapsto \mathbb{R}_+$ des amplitudes locales de fondu :

$$\forall p \in \Gamma, \quad \sigma(p) = \rho \times \frac{\sum_{r \in R} w_b(p, r)}{\max_{q \in \Gamma} \sum_{r \in R} w_b(q, r)}$$

où $w_b(p, r) = e^{-\frac{\|p-r\|^2}{(3\rho\mathcal{R}(r))^2}}$ et ρ est un paramètre utilisateur qui définit la largeur maximale du mélange spatial de patches dans J . En utilisant $\sigma = \sigma(p)$ dans l’équation de mélange de

patches (1), on est capable de faire varier localement l’amplitude du fondu en fonction de la présence ou non d’artéfacts visuels. Par souci de performances, ce calcul s’effectue uniquement sur un sous-domaine $\Omega' \subset \Gamma$ où $\sigma(p)$ est suffisamment élevé.

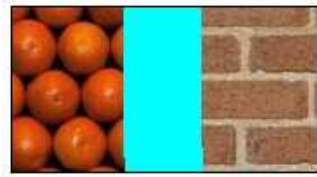
À noter que dans le cas de zones fortement texturées, il peut arriver que 1. et 2. soient vérifiées simultanément sur des points où aucun artéfact visuel n’est présent. Nos expériences suggèrent que réaliser un fondu spatial dans ces zones ne dégrade généralement pas la qualité de la reconstruction.

4 Résultats et Conclusion

La Fig. 3 illustre clairement l’intérêt de notre algorithme de fondu spatial de patches pour l’inpainting. Les artéfacts visuels dûs à des collages de patches incohérents se réduisent fortement, alors que la complexité induite en temps de calcul reste négligeable par rapport à celle des méthodes d’inpainting utilisées. L’effet est proche des résultats de [1, 10], mais notre formule de fondu est directe à calculer (aucune Equation de Poisson à résoudre), et permet plus de flexibilité dans le choix de l’amplitude locale de fondu. Cela rend nos contributions pertinentes pour l’amélioration d’algorithmes d’inpainting génériques utilisant une reconstruction par patch (en y ajoutant seulement deux paramètres utilisateurs τ et ρ , intuitifs et facile à ajuster).

Références

- [1] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro. A variational framework for exemplar-based image inpainting. *IJCV*, 93(3) :319–347, July 2011.
- [2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. of the 27th annual SIGGRAPH conference*, SIGGRAPH ’00, pages 417–424, New York, NY, USA, 2000.
- [3] V. Caselles. Exemplar-based image inpainting and applications. *SIAM News*, 44(10), Decembre 2011.
- [4] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Im. Proc.*, 13(9) :1200–1212, September 2004.
- [5] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Proc. of ICCV-Volume 2*, ICCV ’99, pages 1033–1038, Washington, DC, USA, 1999. IEEE Computer Society.
- [6] P. Harrison. *Image Texture Tools*. PhD thesis, Monash University, 2005.
- [7] N. Kawai, T. Sato, and N. Yokoya. Image inpainting considering brightness change and spatial locality of textures and its evaluation. In *Proc. of the 3rd PSIVT*, PSIVT ’09, pages 271–282, Berlin, Heidelberg, 2009.
- [8] O. Le Meur, J. Gautier, and C. Guillelot. Exemplar-based inpainting based on local geometry. In *ICIP*, pages 3401–3404, Belgique, 2011.
- [9] S. Masnou and J-M. Morel. Level lines based disocclusion. In *ICIP (3)*, pages 259–263, 1998.
- [10] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Transactions on Graphics (SIGGRAPH’03)*, 22(3) :313–318, 2003.
- [11] P. Pérez, M. Gangnet, and A. Blake. Patchworks : Example-based region tiling for image editing. Technical report, Microsoft Research, MSR-TR-2004-04, 2004.
- [12] J. Sun, L. Yuan, J. Jia, and H-Y. Shum. Image completion with structure propagation. *ACM Trans. Graph.*, 24(3) :861–868, July 2005.
- [13] David Tschumperlér and Rachid Deriche. Vector-valued image regularization with pdes : A common framework for different applications. *IEEE Trans. PAMI*, 27(4) :506–517, 2005.
- [14] Mehmet Türkan and Christine Guillelot. Image prediction based on neighbor-embedding methods. *IEEE Trans. Im. Proc.*, 21(4) :1885–1898, 2012.
- [15] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3) :463–476, March 2007.



(a) Image couleur masquée.



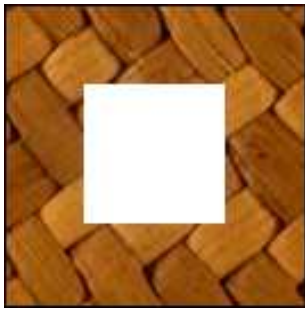
(b) Résultat d'inpainting avec [13].



(c) Résultat d'inpainting avec [4].



(d) Résultat d'inpainting + fondu de patches (algorithme proposé).



(e) Image couleur masquée.



(f) Resultat d'inpainting tiré de [11].



(g) Résultat d'inpainting + fondu de patches (algorithme proposé dans ce papier).



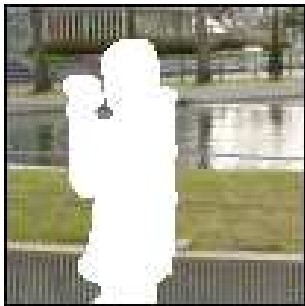
(h) Image couleur masquée.



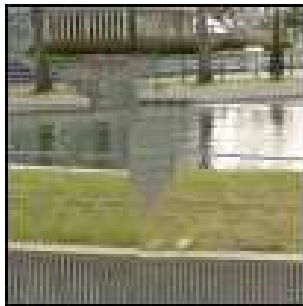
(i) Inpainting obtenu avec [4].



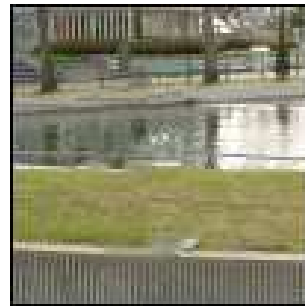
(j) Résultat d'inpainting + fondu de patches (algorithme proposé dans ce papier).



(k) Image couleur masquée (utilisée dans [8]).



(l) Résultat d'inpainting tiré de [8].



(m) Résultat d'inpainting + fondu de patches (algorithme proposé dans ce papier).

FIGURE 3 – Comparaison des résultats de fondu de patches avec des méthodes d'inpainting de l'état de l'art [4, 13, 8, 11].