Les cahiers du GREYC

Année 2013 numéro 1

Steve Macao, David Tschumperlé

Un modèle tensoriel pour le guidage d'inpainting basé patch

Groupe de **R**echerche en **I**nformatique, **I**mage, **I**nstrumentation de **C**aen.
CNRS - UMR 6072

Université de Caen - Campus II
Bd du Maréchal Juin, 14032 Caen Cedex - FRANCE
Tél : 02 31 56 73 31 - Fax : 02 31 56 73 30
E-mail : greyc@info.unicaen.fr

Ecole Nationale Supérieure d'Ingénieur de Caen
Bd du Maréchal Juin, 14050 Caen Cedex - FRANCE
Tél : 02 31 45 25 04 - Fax : 02 31 45 26 98
E-mail : greyc@ensicaen.fr

Introduction

Nous nous intéressons au problème de la reconstruction de données manquantes étendues dans une image couleur. Ce processus, couramment nommé “inpainting”, nécessite une interpolation spatiale intelligente des pixels connus de l’image.

De manière générique, ce terme désigne le fait de déterminer, de la manière la plus automatique possible, la couleur de pixels considérés comme manquants dans une image, c’est-à-dire, dont on ne connait pas les valeurs a-priori. Ce type d’algorithme de reconstruction de données est intéressant à plus d’un titre, de par les nombreuses applications concrètes qu’il peut traiter. Jusqu'à présent, il était très courant qu'un algorithme d’inpainting soit utilisé par exemple pour restaurer de manière numérique des images dégradées par des artefacts ayant détruit de manière complète certaines parties des images (rayures sur des films anciens ou taches sur des photographies). L'application qui nous intéresse dans cette étude consiste en la reconstruction de zones étendues de l'image. C'est le cas des applications de « gomme magique » qui doivent permettre de reconstituer des zones marquées par l'utilisateur tout en conservant les structures cohérentes de l'image. Il y a évidemment de nombreuses autres applications de ce type de technique comme la Réalité Diminuée, ou bien encore la post-production cinématographique.

Cette grande variété d’application explique l’engouement important pour ce type de méthode. On peut tout d’abord citer les travaux précurseurs de Masnou et Morel [20], qui ont été rapidement suivis par une classe d’approches se basant sur les méthodes variationnelles et les EDP de diffusion [4, 6, 12, 27, 28, 30]. Ces méthodes se basent sur les propriétés de lissage non-linéaire des EDP pour diffuser, dans les régions à reconstruire, les valeurs des pixels voisins de ces régions. Leur intérêt principal réside dans leur capacité à reconstruire des données images ayant des géométries complexes (structures non linéaires) à l’intérieur des régions de données manquantes. Cependant, ces méthodes consistent en des lissages ou des transports locaux orientés des intensités des pixels et sont donc incapables de reconstruire des régions texturées, laissant parfois une impression d’aplat non réaliste sur les images traitées qui contiennent initialement beaucoup de textures.

La prise en compte de la texture dans les algorithmes d’inpainting a été envisagée plus récemment sous plusieurs angles. Un formalisme EDP d’inpainting prenant en compte une modélisation des textures a été proposée dans [7], se basant sur une décomposition de l’image dans les espaces BV (fonctions à variations bornées) et G (fonctions oscillantes [21]). Cependant, les textures de grande échelle (macro-textures) sont difficilement reconstruites par ce type d’approche. A l’inverse, dans [2, 14, 16, 32, 35], des algorithmes de synthèse de textures, basées sur des mises en correspondance et des recopies de blocs images sont proposées. Les grandes textures sont alors bien recomposées, mais on perd la propriété de reconstruction d’une géométrie globale cohérente à l’intérieur des régions de données manquantes, notamment quand la région à reconstruire recouvre deux zones de textures différentes. De plus, ces algorithmes (excepté [14]) nécessitent la connaissance d’une texture « modèle » à appliquer, qui n’est pas connue dans le cas général de l’inpainting, où l’image donnée en entrée est constituée en général d’un panel de textures aux caractéristiques très différentes.

Toutefois, il existe une famille de méthode qui a commencé à donner des résultats encourageants dans le cadre général des problématiques d'inpainting. Il s'agit des méthodes basées patch qui traitent la reconstruction de données manquantes comme un puzzle. Et tout particulièrement, celles basées motif (la méthode de Criminisi et al. [1]), qui se révèlent être les plus performantes.

Dans cette étude, nous nous proposons de partir de la méthode de reconstruction basée motif de Criminis et al. et de montrer dans quelle mesure cette approche est très pertinente dans certains cas, mais perd définitivement son efficacité dès lors que l'on souhaite travailler dans un cadre général des applications d'inpainting comprenant des zones manquantes étendues. C'est dans ce contexte que nous proposons des améliorations majeures permettant de résoudre en partie les défauts principaux de la méthode de Criminisi et al. et de permettre la mise en œuvre de solutions de reconstructions satisfaisantes visuellement et fonctionnant dans le cadre général précédemment indiqué.

Cette nouvelle approche s'appuie sur des techniques d'interpolation tensorielle des zones manquantes et de recherche de patch source à l'aide de lignes de champs tensorielles. Nous détaillerons comment nous avons amélioré l'algorithme basé motif de Crimini pour répondre à la problématique générique de reconstitution de données manquantes étendues. Notre algorithme sera finalement illustrée par des résultats sur des données réelles.

1 Méthodes basées motifs

Un des algorithmes les plus influents dans le domaine de l'inpainting basé motifs est celui développé par Criminisi et al. [1] qui consiste à remplir les zones non connues par des patchs connus de l'image en commençant par remplir les zones du contour dans un ordre qui respecte les contraintes de structure de l'image. Nous verrons en détail cet algorithme qui constitue le point de départ du nouvel algorithme proposé s'appuyant sur les tenseurs de structure de l'image.

1.1 L'algorithme de Criminisi et al.

Nous disposons d'une image sur laquelle il existe des zones manquantes à reconstruire. Nous noterons Φ la région source, la zone connue de l'image, Ω la région cible, la zone non connue, et enfin δ la frontière intérieure de Ω (ensemble des pixels de Ω ayant au moins un voisin 8-connexe avec Φ) (illustration 1a). Toutes les illustrations de cette partie sont issues de [1].

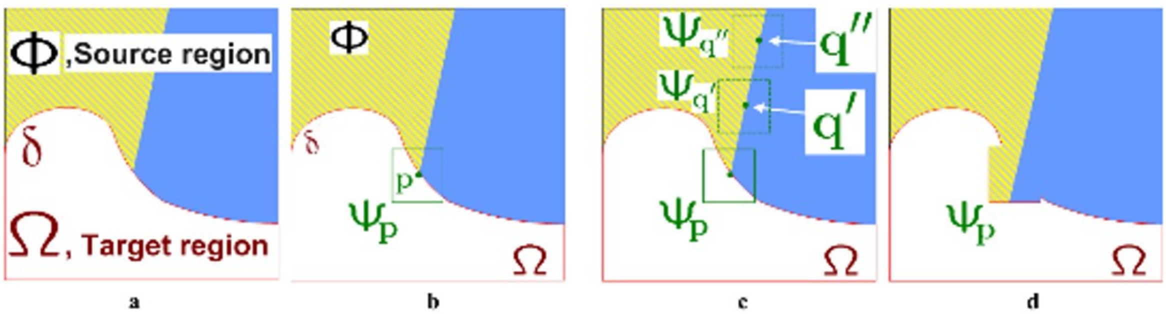


Illustration 1: Reconstruction basée motif

L'objectif est de compléter l'ensemble Ω en y ajoutant des structures géométriques et des textures issues de l'ensemble Φ. Nous utiliserons les pixels contenus dans une zone de forme carrée et centrée sur le pixel p. Ces pixels forment un patch noté Ψ_p (illustration 1b).

Pour chaque point p de δ, une priorité $P(p)$ définit l'ordre de remplissage du patch Ψ_p :

$$P(p)=C(p).D(p)$$

où $C(p)$ est le terme de confiance, et $D(p)$ le terme de données

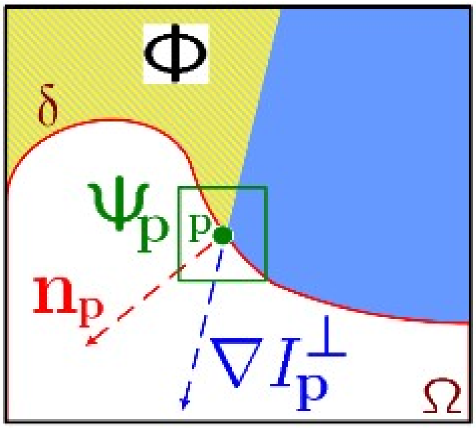


Illustration 2: Informations pour le calcul de la priorité

C(p) correspond à la mesure de la quantité d'information connue autour du pixel *p*. L'objectif étant de compléter en premier les patchs ayant le plus d'information connue. Cela implique que les zones étroites de Ω (en vert, illustration 3a) aient un terme de confiance élevé et tendent à être remplies avant les autres (en rouge, illustration 21a). Ce terme vaut :

$$C(p) = \frac{\sum_{q \in \Psi p \cap \Phi} C(q)}{|\Psi_p|}$$

où $|\Psi_p|$ est le nombre de pixels contenus dans le patch Ψ_p. Ce terme est initialisé à 0 pour tous les pixels *p* de Ω , et à 1 pour ceux de Φ.

D(p) est un terme qui dépend de la quantité d'isophotes débouchant sur le contour δ. Ce terme accroît fortement la priorité des patchs contenant ces isophotes. Par conséquent, la reconstruction des structures géométriques de l'image (en vert, illustration 3b) est priorisée par rapport à celle des zones homogènes, ce qui constitue un point fondamental de l'algorithme. Ce terme est définit par :

$$D(p) = \frac{|\nabla I_p^{\perp} \cdot n_p|}{\alpha} \quad .$$

où α est un facteur de normalisation (255 pour une image typique en niveaux de gris), n_p est un vecteur unitaire, orthogonal à δ au point *p*, et $\nabla I_p^{\perp}$ correspond au gradient maximum dans $\Psi_p \cap \Phi$.

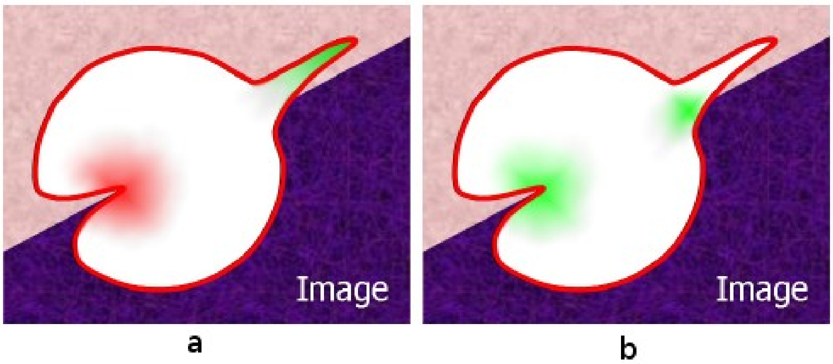


Illustration 3: Priorités données par : a) C(p), b) D(p)

Le pixel ayant la plus forte valeur de *P(p)* devient le pixel prioritaire, et le patch correspondant est rempli. Le patch source Ψ_i est recherché dans la zone Φ avec comme contrainte qu'il soit entièrement inclus dans Φ, et qu'il soit le plus similaire au patch Ψ_p. La comparaison s'effectuant uniquement avec les pixels connus de Ψ_p (illustration 1c).

Le patch Ψ_i résultant minimise la SSD (Sum of Squared Differences). Le patch Ψ_p est alors reconstitué à partir des pixels de Ψ_i (illustration 1d). Puis le contour δ est enfin mis à jour. Les mêmes étapes sont réitérées jusqu'au remplissage complet de Ω.

Les illustrations suivantes donnent un aperçu des résultats obtenus par cet algorithme (illustrations 4 à 7) :

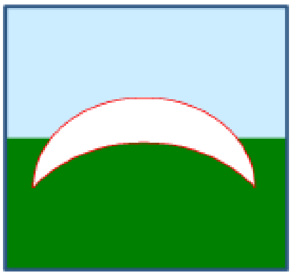


Illustration 4: Image à reconstruire

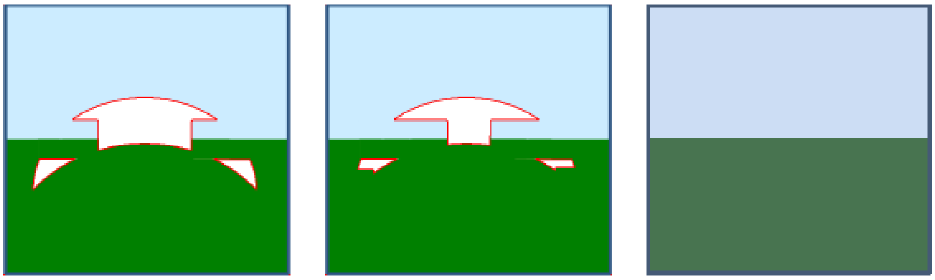


Illustration 5: Quelques itérations de la reconstruction et le résultat final

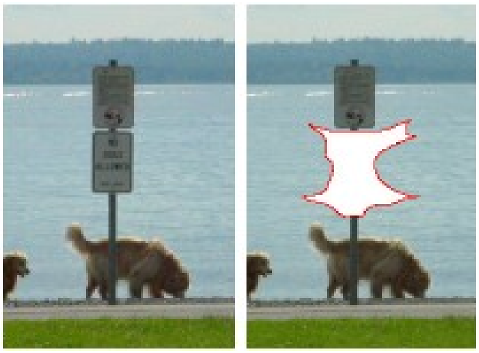


Illustration 6: Images originale et à reconstruire

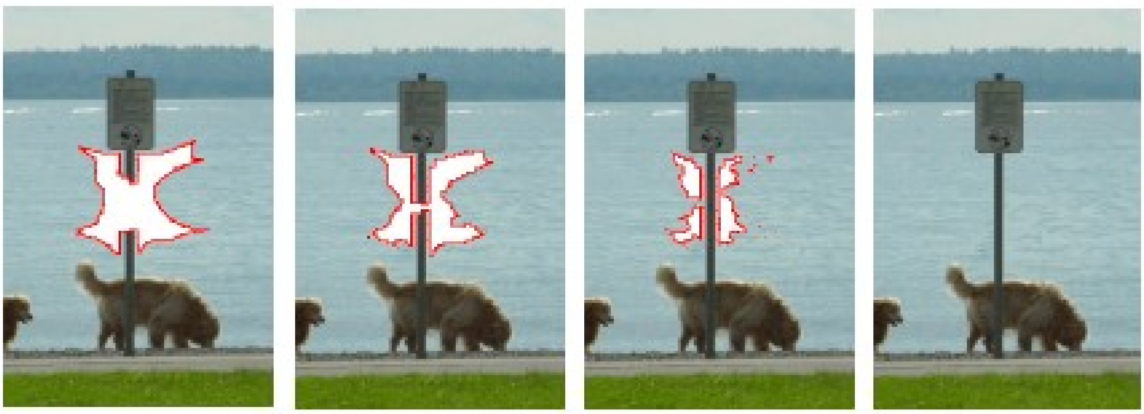


Illustration 7: Quelques itérations et l'image reconstruite

Les structures géométriques (la ligne horizontale, illustration 4, et le poteau de la pancarte, illustration 7) sont bien reconstruites en priorité. Deux autres exemples, où les personnes ont été supprimées de l'image, sont présentés sur l'illustration 8.

Ainsi, l'algorithme de Criminis et al. permet à la fois aux isophotes et aux textures d'être propagés dans Ω. La recopie brute de patchs sources empêche l'effet de flou que l'on obtient avec les méthodes variationnelles (les EDPs par exemple). De plus, il est aussi performant que les techniques existantes pour la reconstruction de petites zones, et est en revanche beaucoup plus efficace dans le cas de larges zones manquantes.

Illustration 8: D'autres reconstructions

Cependant, en y regardant de plus près, l'algorithme donne certes de bons résultats dans les cas où les structures géométriques à reconstruire sont simples et ne se croisent pas, comme celles illustrées dans les images précédentes, mais génère des anomalies visuelles très importantes dès lors que l'on se trouve dans des cas plus usuels d'images ayant des structures plus complexes comme dans les illustrations 9 ci-après.

Image originale	Image à reconstruire	Criminisi et al.

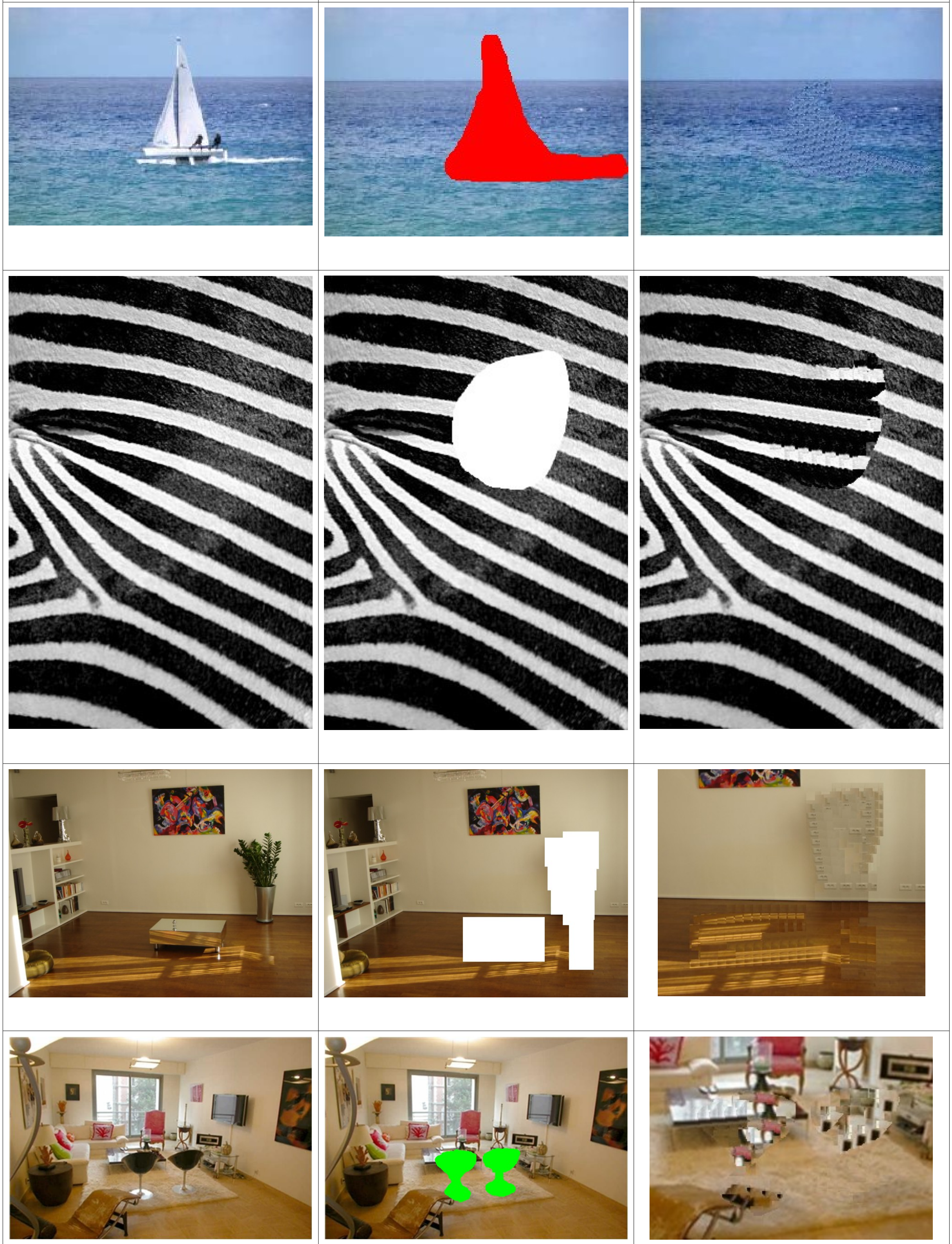


Illustration 9: Défauts de reconstruction des textures

Ces résultats illustrent bien les problèmes liés à la définition du terme de priorité et à la recherche de patch source pour le remplissage. Le terme de priorité repose sur le calcul du plus fort gradient tout au long du remplissage sans avoir une idée des structures qui traversent la zone à remplir. Ce qui induit des erreurs de reconstruction visuellement incohérentes dans le cas de structures qui se croisent ou qui sont courbées. De plus la recopie directe du patch source pose aussi des problèmes de cohérence visuelle, dès lors que l'on se situe plus au centre de la zone manquante ou que des zones de textures se chevauchent.

2 Nouvelle approche par interpolation tensorielle des zones manquantes

Dans cette nouvelle approche, nous traitons le cas des limitations importantes de l'algorithme de Criminisi et al. Comme cela a été indiqué, cela provient principalement du fait que la reconstitution des zones manquantes se fait uniquement sur une analyse locale des structures de l'image au niveau des contours des zones non connues. Cette approche locale ne permet pas d'avoir une estimation globale des structures de l'image nécessaires à une reconstitution visuelle cohérente et de qualité. Aussi nous nous proposons de réaliser, avant l'étape de reconstruction, une estimation des structures des zones manquantes par interpolation des tenseurs de structure.

2.1 Interpolation des contours à partir des tenseurs de structures

L'estimation des tenseurs à l'intérieur de Ω est réalisée par interpolation des tenseurs de Φ. L'objectif est d'obtenir un « mapping » global des structures de Ω avant reconstruction de l'image. Cette interpolation permettra de mettre en place une nouvelle formule de priorité pour l'ordre de remplissage de Ω.

2.1.1 Tenseur de structures

Sur une image scalaire classique $I : E \rightarrow \mathbb{R}$, la direction des contours est donnée par la direction orthogonale au gradient. Pour les images couleurs (et multi-valuées en général), cette notion s'étend par l'utilisation du tenseur de structures. Pour plus de robustesse dans l'estimation de la géométrie locale des images, il est fréquent de lisser d'une part les différents gradients ∇R, ∇G, ∇B avant le calcul du champ de tenseurs de structures, et d'autre part le champ lui-même après calcul.
Pour une image multi-valuée $I: E \rightarrow \mathbb{R}^n$, le champ de tenseurs de structures se définit comme l'image $G : E \rightarrow P(2)$ dont les valeurs sont les matrices 2x2 symétriques et définies positives suivantes :

$$\forall (x,y) \in E,\ G(x,y) = \sum_{k=1}^{n} \nabla I_k(x,y) \cdot \nabla I_k(x,y)^T$$

Dans le cas d'images couleur, n=3, et k représente les composantes R, G et B. L'expression de ces tenseurs de structures pour des images couleur en fonction des dérivées premières R_x, R_y, G_x, G_y, B_x, B_y des composantes (R, G, B) par rapport à x et y, est :

$$G(x,y) = \begin{bmatrix} R_x^2 + G_x^2 + B_x^2 & R_x R_y + G_x G_y + B_x B_y \\ R_x R_y + G_x G_y + B_x B_y & R_y^2 + G_y^2 + B_y^2 \end{bmatrix}$$

avec $R_x = \frac{\partial I_R}{\partial x} = \frac{I_R(x+1,y) - I_R(x-1,y)}{2}$ et $R_y = \frac{\partial I_R}{\partial y} = \frac{I_R(x,y+1) - I_R(x,y-1)}{2}$ par exemple

Un tenseur de structure possède deux valeurs propres λ_1 et λ_2 positives associées à deux vecteurs propres v_1 et v_2. Ces vecteurs propres sont orthogonaux, l'un se trouve dans la direction du gradient, l'autre dans la direction de la tangente au contour au point (x, y). Graphiquement, un tenseur de structure se représente sous la forme d'une ellipse dont le plus grand rayon est orienté selon le gradient le plus fort et a pour valeur la plus grande valeur propre, et dont le petit rayon est tangent au contour et de valeur la plus petite valeur propre (illustration 10).

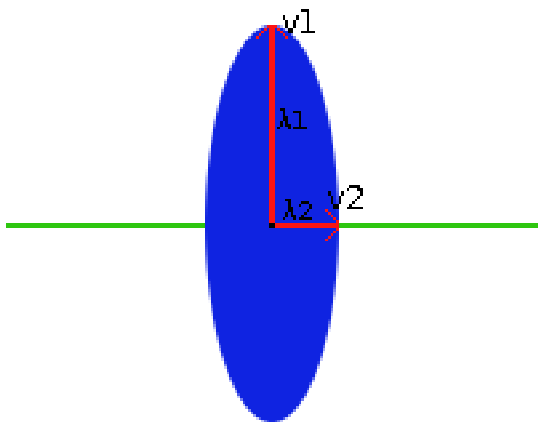


Illustration 10: Tenseur de structure en 2D

Un tenseur est dit « anisotrope » s'il correspond à un pixel de contour, il est dit « isotrope » s'il correspond à un pixel de zone homogène.

Ces tenseurs sont évalués pour chaque point de Φ. Toutefois, nous excluons de Φ les points ayant au moins un voisin 8-connexe avec Ω. La frontière $\delta 2$ désigne l'ensemble des points de Φ ayant au moins un voisin 8-connexe dans Ω, soit la frontière extérieure de Ω, et la frontière $\delta 1$ désigne l'ensemble des points de Ω, n'appartenant pas à $\delta 2$, ayant au moins un voisin 8-connexe dans $\delta 2$ (illustration 11), soit la frontière intérieure de Ω.

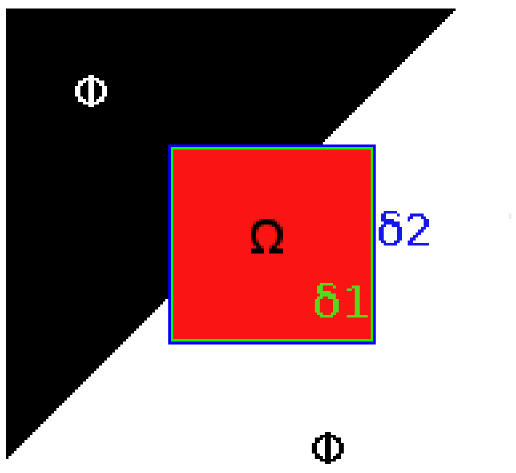
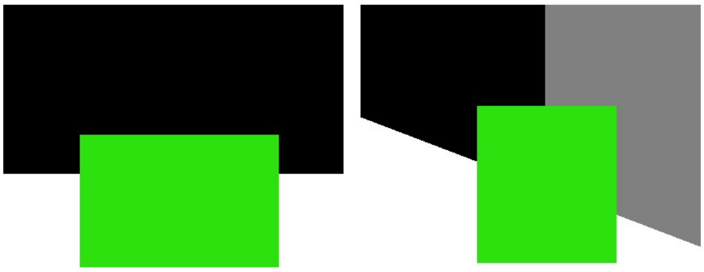


Illustration 11: Ensembles Φ et Ω, frontières $\delta 1$ et $\delta 2$

2.1.2 L'interpolation des tenseurs de structures

Plusieurs types d'interpolation sont possibles, mais nous nous sommes focalisés dans une premier temps sur les méthodes d'interpolation linéaire des tenseurs puis, dans un deuxième temps, sur les méthodes d'érosion tensorielle.

Les images utilisées pour les tests d'interpolation sont celles de l'illustration 12. Elles ont été choisies car elles représentent un cas simple (illustration 12a) et un cas plus complexe (illustration 12b) de structures à reconstruire.



a b

Illustration 12: a) Ligne simple horizontale,
b) Jonction de trois lignes

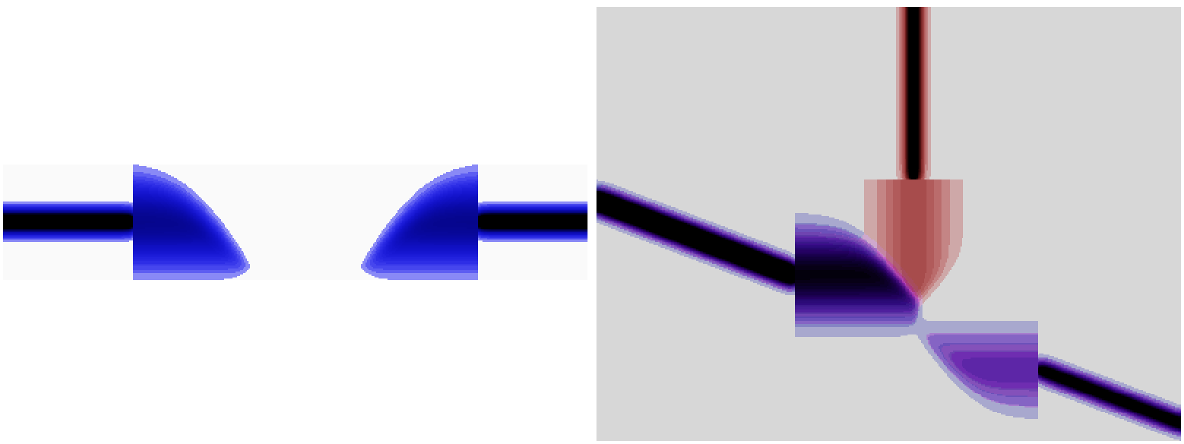
2.1.2.1 Interpolation linéaire

L'interpolation linéaire consiste à donner pour valeur à chaque point *p* de *Ω*, une moyenne des points *q* du contour *δ2*, et plus précisément une moyenne des valeurs des tenseurs en ces points, pondérée par la distance euclidienne entre les points. La valeur prise par un point *p* de Ω est donnée par :

$$G(p) = \frac{\sum_{q \in \delta 2} w(p,q) \cdot G(q)}{\sum_{q \in \delta 2} w(p,q)} \quad \text{avec} \quad w(p,q) = \exp^{\frac{-d(p,q)^2}{2\sigma^2}}$$

avec σ une constante réelle strictement positive et *d* est la distance euclidienne.

Ainsi, plus les points sont proches, plus le poids est important, et plus le tenseur correspondant a de l'influence dans la moyenne.



a b

Illustration 13: Interpolation linéaire

Les structures reconstruites sont assez peu précises dans la mesure où aucune notion géométrique n'intervient dans cette reconstruction, mais seulement la distance aux points de

$\delta 2$. Or un point de Ω peut très bien faire partie d'une structure géométrique, sans pour autant être proche du point de $\delta 2$ appartenant à cette structure. C'est le cas sur l'illustration 13a, où les points censés prolonger la ligne horizontale se retrouvent plus proches des points de la zone homogène de couleur noire que de ceux de la frontière entre les zones noire et blanche. L'interpolation est donc erronée en ces points, et la ligne n'est pas correctement reconstruite.

2.1.2.2 Erosion tensorielle simple

L'interpolation linéaire s'est révélée être trop imprécise, et nous a amené à nous intéresser aux méthodes non linéaires comme l'érosion tensorielle.

Chaque valeur du tenseur prise au point de $\delta 2$, dans l'ordre de parcours de l'image, est recopié sur ses voisins 8-connexes appartenant à $\delta 1$. Tous les points de $\delta 2$ parcourus, les points de $\delta 1$ complétés deviennent le contour $\delta 2$, et une nouvelle frontière $\delta 1$ est définie. La recopie des valeurs de $\delta 2$ sur $\delta 1$ est réitérée jusqu'au remplissage complet de Ω.

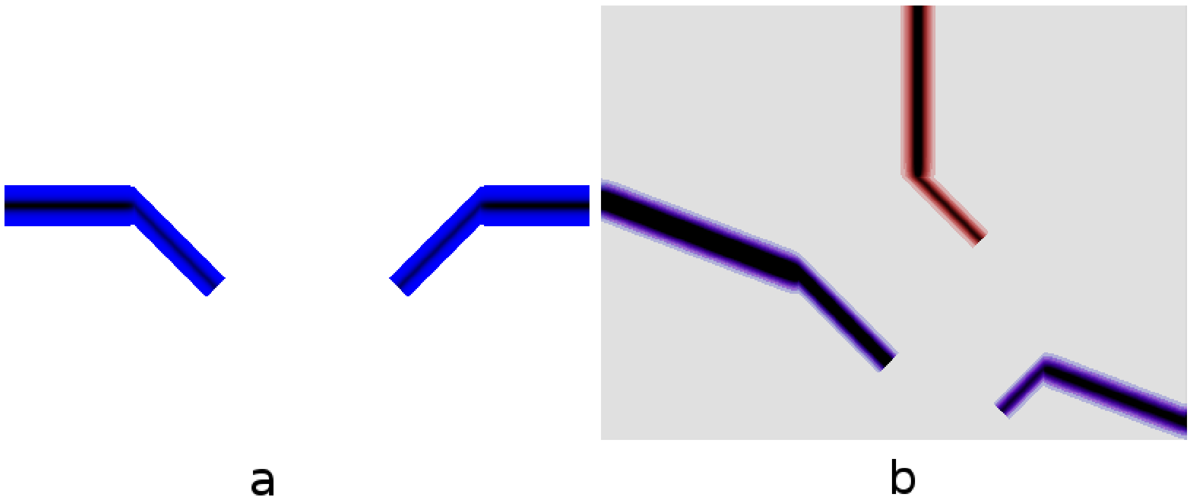


Illustration 14: Interpolation tensorielle simple

Dans le cas de nos images test, l'érosion agit en premier sur des pixels de zones homogènes, ce qui induit un biais dans la recopie des pixels de structures (illustration 14). La limite de cette approche réside dans la non prise en compte de la direction des structures dans la recopie.

2.1.2.3 Erosion tensorielle : approche hybride

Dans cette approche, notre objectif est de prendre en compte la direction des structures lors de l'érosion tensorielle. La direction des structures du contour est donnée par les vecteurs propres du tenseur de structure. L'idée est alors de combiner l'approche de l'interpolation linéaire et de l'érosion tensorielle en intégrant la connaissance sur les structures des contours. C'est en cela qu'il s'agit d'une approche hybride.

En lieu et place de la recopie directe des valeurs des tenseurs de $\delta 2$ sur $\delta 1$, une interpolation linéaire pondérée est réalisée des valeurs des tenseurs de $\delta 2$ pour chaque pixel de $\delta 1$. La pondération est pour sa part fonction de la direction des tenseurs du contour :

$$\forall p \in \delta_1,\ G(p) = \frac{\sum_{q \in \delta 2} w_2(p,q) \cdot G(q)}{\sum_{q \in \delta 2} w_2(p,q)} \quad \text{avec} \quad w_2(p,q) = \exp^{\frac{-u^T \cdot G_1(p) \cdot u}{2\sigma^2}}$$

σ est une constante réelle strictement positive et u est le vecteur $\vec{pq}$. $G_1(p)$ est la valeur du tenseur de structure interpolé calculée dans une étape préalable :

$$\forall p \in \delta_1,\ G_1(p) = \frac{\sum_{q \in \delta 2} w_1(p,q) \cdot G(q)}{\sum_{q \in \delta 2} w_1(p,q)} \quad \text{avec} \quad w_1(p,q) = \exp^{\frac{-u^2}{2\sigma^2}}$$

Ce nouveau poids permet de mieux prendre en compte la direction des tenseurs de δ2 en leur donnant plus de poids selon la direction du vecteur $\vec{pq}$. En effet, G_1 peut se décomposer sous la forme $G_1 = \lambda_1 v_1 v_1^t + \lambda_2 v_2 v_2^t$, ainsi :

$$u^T \cdot G_1 \cdot u = \lambda_1 \cdot (u^T \cdot v_1) \cdot (v_1^T \cdot u) + \lambda_2 \cdot (u^T \cdot v_2) \cdot (v_2^T \cdot u)$$

Par conséquent, dans le cas où G_1 est anisotrope :

- si u tend vers un vecteur colinéaire à v_1 , soit orthogonal à la structure géométrique, alors le premier terme de l'addition tend vers la valeur $\lambda_1 * ||u|| * ||v_1||$ (importante car G_1 est anisotrope), le deuxième tend vers 0, et $w_2(p, q)$ tend vers une petite valeur. Les points situés dans la direction orthogonale à la structure ont donc un poids faible dans le calcul de $G(p)$.
- si u tend vers un vecteur colinéaire à v_2 , soit tangent à la structure, alors le premier terme tend vers 0, le deuxième tend vers la valeur $\lambda_2 * ||u|| * ||v_2||$ (faible car G_1 est anisotrope), et $w_2(p, q)$ tend vers 1. Les points situés dans la direction tangente à la structure ont donc un poids élevé dans le calcul de $G(p)$.

Dans le cas où G_1 est isotrope, les deux termes de l'addition sont proches, et donc les points situés dans les directions orthogonale et tangente à la structure ont des poids à peu près équivalents.

Il est à noter que dans le cas de zones manquantes étendues, il est préférable de ne pas recalculer la nouvelle frontière $\delta 2$ à chaque itération et de conserver la frontière initiale $\delta 2_0$ comme tenseur de référence. De cette manière, les tenseurs de structures calculés dans Ω deviennent de plus en plus isotropes à mesure que l'on avance vers l'intérieur de Ω. L'incertitude sur les structures géométriques s'accroît alors quand on s'éloigne de la frontière $\delta 2_0$ initiale qui contient l'information sur les structures à interpoler.

Au final, on récupère bien pour le point p un tenseur dont la direction se rapproche le plus possible de celle de la structure géométrique à laquelle il pourrait appartenir. On constate une nette amélioration du résultat sur l'illustration 15b.

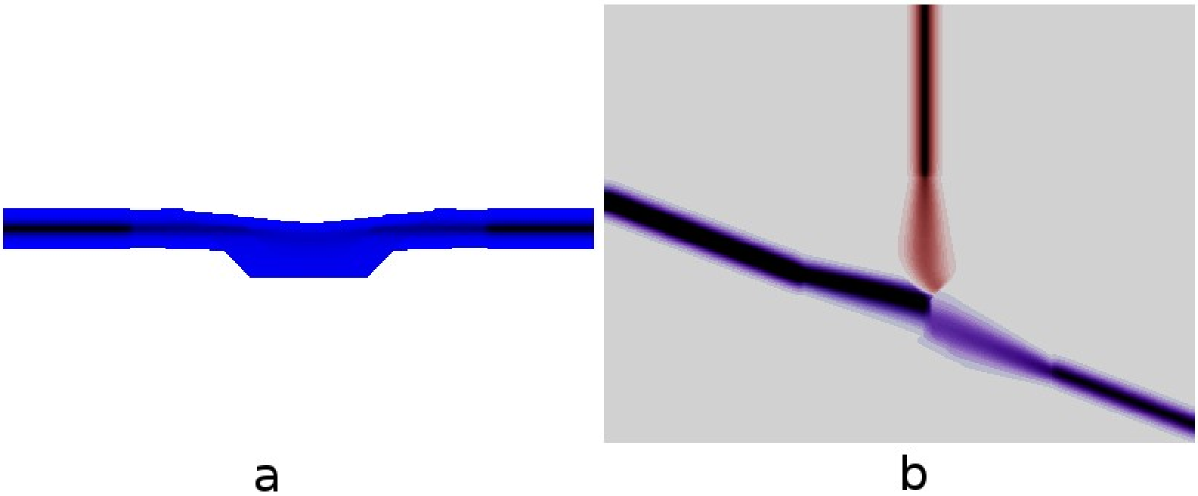


Illustration 15: Erosion tensorielle avec moyennes selon les distances et les valeurs des tenseurs

2.2 Facteur anisotrope

L'intérêt premier de réaliser une interpolation des tenseurs de structures est la mise en place d'une nouvelle priorité pour l'ordre de remplissage de Ω : le facteur anisotrope *FA*. Ce terme permet de mesurer l'anisotropie d'un tenseur et donc de déterminer si ce dernier correspond d'avantage à une structure géométrique ou à une zone homogène. Le facteur anisotrope *FA* se calcule soit par la différence des valeurs propres du tenseur interpolé, soit par leur rapport:

$$FA=\lambda_1-\lambda_2 \;\; ou \;\; FA=\frac{\lambda_1}{\lambda_2+\epsilon}$$

avec λ_1 est la plus grande valeur propre, λ_2 la plus petite, et ϵ une constante strictement positive entre 0 et 1 (pour éviter la division par zéro). La valeur du *FA* se calcule en chacun des points de la zone interpolée Ω.

L'interpolation des tenseurs de structure nous a permis de créer dans Ω des structures géométriques proches de celles que l'on espère obtenir lors de la reconstruction de l'image. Par conséquent, les points correspondant à des structures géométriques ont un *FA* important contrairement aux points de zones homogènes. Ainsi, le terme de priorité de l'algorithme de Criminisi et Al. devient :

$$\forall p\in\Omega \;,\; P(p)=FA(p)$$

Il est à noter que, dans le cas où la frontière initiale $\delta 2_0$ est prise comme référence tout au long de la reconstruction, les priorités sont fortes sur les bords de Ω, et diminuent de plus en plus en s'enfonçant dans Ω. Cela a pour effet (illustrations 16 et 17) de favoriser la reconstruction de toutes les structures géométriques avant les zones homogènes en fonction de leur anisotropie :

- contrairement à la priorité de Criminisi et al., la structure dominante ne se prolonge pas dans tout Ω,
- il est inutile d'effectuer un remplissage purement concentrique pour reconstruire les différentes structures.

Ainsi, les priorités basées sur le facteur anisotrope ne sont pas mises à jour après chaque complétion de patch. Elles sont fixées pour toute la reconstruction à partir de l'interpolation des tenseurs de structure de l'image initiale.

Illustration 16: Priorité donnée par FA = $\lambda_1 - \lambda_2$

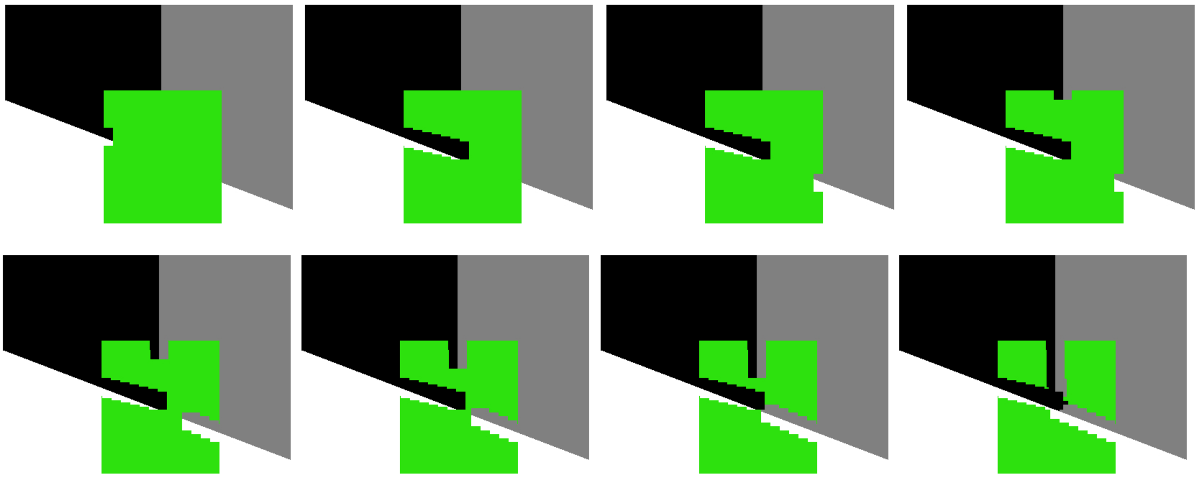


Illustration 17: Priorité donnée par FA = $\lambda_1 - \lambda_2$

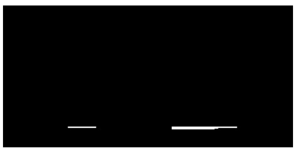


Illustration 18: Résultat de l'illustration 16

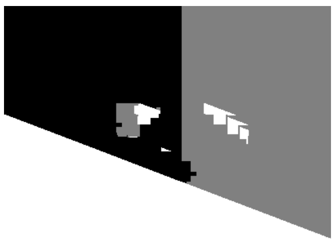


Illustration 19: Résultat de l'illustration 17

2.3 Recherche des patchs sources par calcul des lignes de champs

Le deuxième grand intérêt de l'interpolation tensorielle des zones manquantes est notre capacité à exploiter les lignes de champs issues des tenseurs de structures calculés, y compris dans la zone non connue, pour la recherche des patchs sources lors de la reconstruction.

2.3.1 Calcul des lignes de champs

Les lignes de champs ont initialement été utilisées en imagerie, pour la visualisation de champs de vecteurs (méthodes des LICs) [29]. Elles servent également en débruitage et régularisation d'image [30]. L'idée est d'utiliser ces techniques dans le cadre de l'inpainting basé motifs pour la recherche et la sélection des patchs.

Les lignes de champs de tenseurs de structure se déduisent à partir d'un champ de tenseurs contours (« edge tensor ») définis comme suit :

$$E = \frac{1}{(\lambda_1 + \lambda_2 + \varepsilon)^m} \cdot v_1 \cdot v_1^T + \frac{1}{(\lambda_1 + \lambda_2 + \varepsilon)^n} \cdot v_2 \cdot v_2^T$$

avec m et n deux réels positifs.

L'objectif est de parcourir toutes les directions à partir d'un point du contour tout en restant confiné dans des zones de structures similaires.

Soit le vecteur $u = (\cos \alpha, \sin \alpha)$ avec α qui varie entre 0 et 350 degrés, avec un pas de 10 degrés. Alors le vecteur $E.u$ vérifie :

$$E \cdot u = \frac{1}{(\lambda_1 + \lambda_2 + \varepsilon)^p} \cdot v_1 \cdot v_1^T \cdot u + \frac{1}{(\lambda_1 + \lambda_2 + \varepsilon)^q} \cdot v_2 \cdot v_2^T \cdot u$$

– si E est anisotrope, alors E peut s'écrire approximativement sous la forme $E = \lambda.w.w^T$, avec λ constante réelle strictement positive. Ainsi :

$$E \cdot u = \lambda \cdot w \cdot w^T \cdot u = \lambda \cdot (w^T \cdot u) \cdot w = C \cdot w$$

avec C constante réelle positive.

$E.u$ est alors colinéaire au vecteur w qui donne localement la direction de la structure géométrique. Si u est dans la direction de w, alors C est non nul, et on longe bien la structure. Si, à l'inverse, u est orthogonal à w, alors C est nul, et $E.u = 0$

– si E est isotrope (zone homogène) alors E vaut approximativement $\lambda.Id$. Donc :

$$E \cdot u = \lambda \cdot Id \cdot u = \lambda \cdot u$$

$E.u$ est alors colinéaire au vecteur u dont la direction varie avec l'angle α. On balaye ainsi toutes les directions.

Les images des vecteurs (une pour chaque valeur de α) donnent la direction et le sens de parcours des lignes de champs. A partir du pixel prioritaire, chaque image est traitée séparément pour déduire la ligne de champs qui lui correspond.

Au final, les lignes de champs sont calculées pour chaque direction qui part du pixel prioritaire et se déploient différemment selon la zone où elles se trouvent (illustration 20).

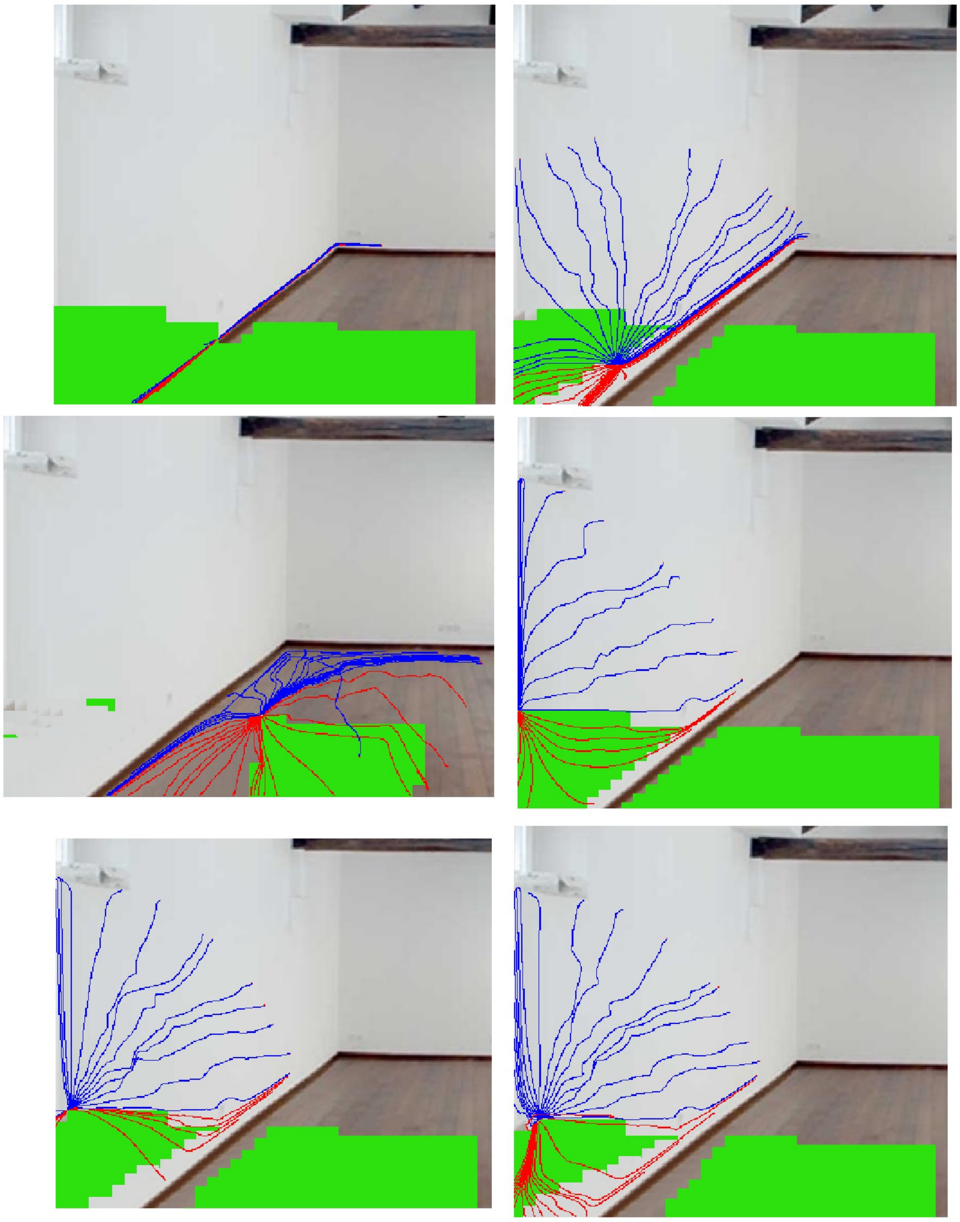


Illustration 20: Lignes de champs sur une image réelle

2.3.2 Recherche des patchs sur les lignes de champs

Les lignes de champs calculées suivent les directions des structures de l'image, que ce soit dans les zones inconnues ou dans les zones connues. Dans le cas de la recherche de patchs source, cette propriété des lignes de champs nous assure que la recherche des patchs ne traversera pas les structures de l'image, et favorisera la recherche dans des zones confinées proches de la structure du pixel de contour.

Cette recherche est traitée de la manière suivante :

- Chaque ligne de champs est parcourue séparément,
- Chaque demi-ligne d'origine le pixel prioritaire récupère le meilleur patch (le centre du patch doit appartenir à la ligne de champs), s'il existe (en effet, une demi-ligne peut être intégralement dans la zone Ω et ne pas posséder de patch source valide). Dans le cas où

aucun patch source n'est valide, une zone de recherche rectangulaire est utilisée comme dans l'algorithme de Criminisi et al. initial.

Il est à noter que dans la mesure où l'interpolation des tenseurs de structure est de plus en plus incertaine vers le centre de Ω, les lignes de champs qui y sont calculées sont de moins en moins précises et l'espace de recherche qui en résulte peut être erronée. Un des moyens de corriger cela est de réaliser une réinterpolation des tenseurs tous les N patchs complétés afin de mettre à jour les structures géométriques en fonction de ce qui a déjà été reconstruit. Précisons que les priorités données par le facteur anisotrope ne sont, pour leur part, pas modifiées.

2.3.3 Moyenne des patchs

La simple recopie des meilleurs patchs dans Ω introduit des artefacts lors de la reconstruction, tout particulièrement dans les zones homogènes. Nous avons dès lors introduit un calcul de moyenne de patchs basée sur l'algorithme de débruitage des Non-local means (NL-means).

Le patch cible n'est donc plus complété uniquement par un seul patch source, mais il est le résultat d'une moyenne pondérée des meilleurs patchs lui ressemblant. Ces patchs étant sélectionnés sur chaque ligne de champs.

En notant Ψ_p le patch cible et Ψ_i un patch source, cela donne :

$$\Psi_p = \frac{\sum_{i=1}^{n} w(\Psi_i) \cdot \Psi_i}{\sum_i w(\Psi_i)} \quad \text{avec} \quad w(\Psi_i) = \exp^{\frac{-SSD(\Psi_p, \Psi_i)}{2\sigma^2}}$$

avec n est le nombre de patchs pris en compte dans la moyenne, et σ une constante réelle strictement positive.

2.4 Lissage final des patchs reconstruits

Le résultat final fait apparaître tout de même un phénomène de pixelisation des zones reconstruites, et plus précisément les frontières des patchs restent visibles (illustration 21a). Afin d'atténuer ce phénomène et de conserver l'unité globale de l'image après la reconstruction, nous effectuons un lissage de l'image finale en associant à chaque pixel de Ω la moyenne de ses voisins 4-connexes (illustration 21b)

a b c

Illustration 21: a) Image à reconstruire, b) Reconstruction sans lissage, c) Reconstruction avec lissage

2.5 Résultats, illustrations

Sur les exemples ci-après, la méthode d'interpolation tensorielle et de recherche de ligne de champs de tenseur a été mise en œuvre avec des résultats bien plus performant que la méthode de Criminisi et Al.

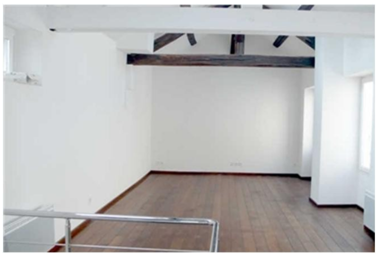
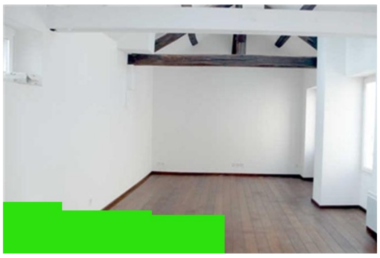
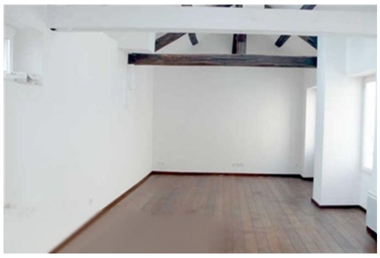
Image originale	Masque	Reconstruction
		
		
	 18	

Illustration 22

Les structures géométriques reconstruites sont satisfaisantes et se rapprochent bien de celles attendues. L'effet de blocs qui apparaissait avec la recopie d'un unique patch, et le mauvais raccord des zones homogènes sont bien atténués.

2.5.1 Synthèse méthode

Le nouvel algorithme de reconstruction basé motif par interpolation tensorielle peut se résumer ainsi :

1. Interpolation des tenseurs de structure dans les zones manquantes par approche hybride (érosion tensorielle et interpolation linéaire)
2. Calcul du facteur anisotropique *FA(p)* pour chaque point de Ω à partir de l'interpolation tensorielle initiale.
3. Répéter jusqu'à que $\Omega = \emptyset$:
   a) Calcul du contour extérieur δ de Ω
   b) Sélectionner le patch cible à partir du point prioritaire de δ ayant le plus grand facteur anisotropique *FA(p)* issu de l'interpolation initiale.
   c) Calcul des lignes de champs partant de ce point prioritaire.
   d) Recherche des meilleurs patchs source à moyenner sur chaque ligne de champs.
   e) Copie de la moyenne des patchs sources dans le patch cible du point prioritaire.
   f) Toutes les N itérations, ré-évaluer une interpolation tensorielle de la zone restante à remplir.
4. Lissage finale de Ω

3 Conclusion

L'algorithme de Criminisi et al. initial a été revu complètement : la formule de priorité a été modifiée, et la recherche des patchs sources a été améliorée. La nouvelle approche de reconstruction par interpolation tensorielle que nous avons proposés est générique et efficace. Les résultats visuels sont crédibles et apportent une vraie alternative aux méthodes proposées jusqu'à présent dans le domaine de l'inpainting.

The authors would like to thank M. Daisy for his suggestions.

Références bibliographiques

[1] « Region Filling and Object Removal by Exemplar-Based Image Inpainting », A. Criminisi, P. Pérez and K. Toyama, Microsoft Research, Cambridge (UK) and Redmond (US).

[2] « Exemplar-based Inpainting with Rotation Invariant Patch Matching », Jiri Boldys, Bernard Besserer, L3i, University of La Rochelle, Av. Michel Crépeau, 17042 La Rochelle cedex 1, France.

[3] « Image Repairing: Robust Image Synthesis by Adaptive ND Tensor Voting », Jiaya Jia and Chi-Keung Tang, Vision and Graphics Group, Computer Science Department, Hong-Kong University of Science and Technology.

[4] « Spatio-temporal Texture Synthesis and Image Inpainting for Video Applications », Sanjeev Kumar, Mainak Biswas, Serge J. Belongie and Truong Q. Nguyen, University of California, San Diego.

[5] « A Survey on Variational Image Inpainting, Texture Synthesis and Image Completion », Isik Baris Fidaner, Bogazici University.

[6] « A Non-local Algorithm for Image Denoising », Antoni Buades, Bartomeu Coll, Dpt. Matematiques i Informatica, UIB, Ctra. Valldemossa Km. 7.5, 07122 Palma de Mallorca, Spain, Jean-Michel Morel, CMLA, ENS Cachan, 61 Av du Président Wilson, 94235 Cachan, France.

[7] « An Image Inpainting Technique Based on the Fast Marching Method », Alexandru Telea, Eindhoven University of Technology.

[8] S. Masnou and J. Morel, « Level Lines Based Disocclusion », in Image Processing, 1998. ICIP 98. Proceedings. 1998 International Conference on.

[9] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, « Image Inpainting », in SIGGRAPH '00 : Proceedings of the 27th annual conference on Computer graphics and interactive techniques. New York, NY, USA : ACM Press / Addison-Wesley Publishing Co., 2000, pp. 417-424.

[10] A. Rares, M. J. T. Reinders, and J. Biemond, « Edge-based Image Restoration ». IEEE Transactions on Image Processing, vol. 14, no. 10, pp. 1454-1468, 2005.

[11] D. J. Heeger and J. R. Bergen, « Pyramid-based texture analysis / synthesis », in SIGGRAPH, 1995, pp. 229-238.

[12] J. Portilla and E. P. Simoncelli, « A Parametric Texture Model Based on Joint Statistics of Complex Wavelet Coefficients », International Journal of Computer Vision, vol. 40, no. 1, pp. 49-70, 2000.

[13] V. Kwatra, I. Essa, A. Bobick and N. Kwatra, « Texture Optimization for Example-based Synthesis », in SIGGRAPH '05 : ACM SIGGRAPH 2005 Papers. New York, NY, USA : ACM Press, 2005, pp. 795-802.

[14] A. Efros and T. Leung, « Texture Synthesis by Non-Parametric Sampling », in ICCV (2), 1999, pp. 1033-1038.

[15] L.-W. Wey and M. Levoy, « Fast Texture Synthesis Using Tree-structured Vector Quantization », in SIGGRAPH 2000, Computer Graphics Proceedings, K. Akeley, Ed. ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000, pp. 479-488.

[16] M. Ashikhmin, « Synthetizing Natural Textures », in SI3D '01 : Proceedings of the 2001 symposium on Interactive 3D graphics. New York, NY, USA : ACM Press, 2001, pp. 217-226.

[17] A. A. Efros and W. T. Freeman, « Image Quilting for Texture Synthesis and Transfer », in SIGGRAPH 2001, Computer Graphics Proceedings, E. Fiume, Ed. ACM Press / ACM SIGGRAPH, 2001, pp. 341-346.

[18] L. Liang, C. Liu, Y.-Q. Xu, B. Guo and H.-Y. Shum, « Real-time Texture Synthesis by Patch-based Sampling », ACM Trans. Graph., vol. 20, no. 3, pp. 127-150, 2001.

[19] L. Tonietto, M. Walter and C. R. Jung, « Patch-based Texture Synthesis Using Wavelets », in SIBGRAPI '05 : Proceedings of the XVIII Brazilian Symposium on Computer Graphics and Image Processing. Washington, DC, USA : IEEE Computer Society, 2005, p. 383.

[20] M. Bertalmio, L. Vese, G. Sapiro and S. Osher, « Simultaneous Structure and Texture Image Inpainting », cvpr, vol. 02, p. 707, 2003.

[21] J. Jia and C.-K. Tang, « Image Repairing : Robust Image Synthesis by Adaptive ND Tensor Voting », cvpr, vol. 01, p. 643, 2003.

[22] I. Drori, D. Cohen-Or and H. Yeshurun, « Fragment-based Image Completion », in SIGGRAPH '03 : ACM SIGGRAPH 2003 Papers. New York, NY, USA : ACM Press, 2003, pp. 303-312.

[23] S. X. D. Nie, L. Ma, « Similarity Based Image Inpainting Method », in Multi-Media Modelling Conference Proceedings, 2006 12th International, 2006, p. 4.

[24] X. Shao, Z. Liu and H. Li, « An Image Inpainting Approach Based on the Poisson Equation », in DIAL '06 : Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06). Washington, DC, USA : IEEE Computer Society, 2006, pp. 368-372.

[25] T. Chan, S.H. Kang and J. Shen, « Euler's Elastica and Curvature Based Inpainting », SIAM Journal of Applied Mathematics, 2002.

[26] T.F. Chan and J. Shen, « Non-texture Inpainting by Curvature-driven Diffusions (CDD) », Journal of Visual Communication and Image Representation, 12(4): 436--449, 2001.

[27] D. Tschumperle and R. Deriche, « Vector-Valued Image Regularization with PDE's : A Common Framework for Different Applications », IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), Vol.27, No,4, 2005.

[28] Méthode Runge-Kutta pour le calcul de streamlines : W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. « Runge-Kutta Method in Numerical Recipes in FORTRAN: The Art of Scientific Computing », Cambridge University Press, pp. 704-716, 1992.

[29] Utilisation initiale des streamlines en imagerie, pour la visualisation de champs de vecteurs (méthode des LICs) : B. Cabral and L.C. Leedom. « Imaging vector fields using line integral convolution », SIGGRAPH'93, in Computer Graphics Vol.27, pp.263--272, 1993.

[30] Débruitage/Régularisation d'image, en utilisant les streamlines : D. Tschumperle, « Fast Anisotropic Smoothing of Multi-Valued Images using Curvature-Preserving PDE's », International Journal of Computer Vision, Vol. 68, No. 1, pp. 65--82, 2006, ISSN : 0920-5691.

[31] Site web officiel de la bibliothèque CImg : http://cimg.sourceforge.net/