

Object Removal by Exemplar-Based Inpainting

Kévin Polisano

A partir d'un article de A. Criminisi, P. Pérez & H. K. Toyama

14/02/2013



- 1 Introduction
- 2 Synthèse de texture
- 3 Interpolation par EDP
- 4 L'union fait la force
- 5 Résultats et comparaison
- 6 Conclusion

Contents

- 1 Introduction
- 2 Synthèse de texture
- 3 Interpolation par EDP
- 4 L'union fait la force
- 5 Résultats et comparaison
- 6 Conclusion

Introduction

Qu'est-ce que l'inpainting ?

- Restauration d'images



Introduction

Qu'est-ce que l'inpainting ?



www.cornell.edu/~enlighten.com



Introduction

Qu'est-ce que l'inpainting ?

- Suppression de texte



Introduction

Qu'est-ce que l'inpainting ?

- Suppression d'objets / large régions dans une image en conservant une information pertinente



Introduction

Qu'est-ce que l'inpainting ?

- Délimiter une zone à reconstruire et y propager l'information extérieure



Introduction

Méthodologie de la restauration

Méthodologie en peinture

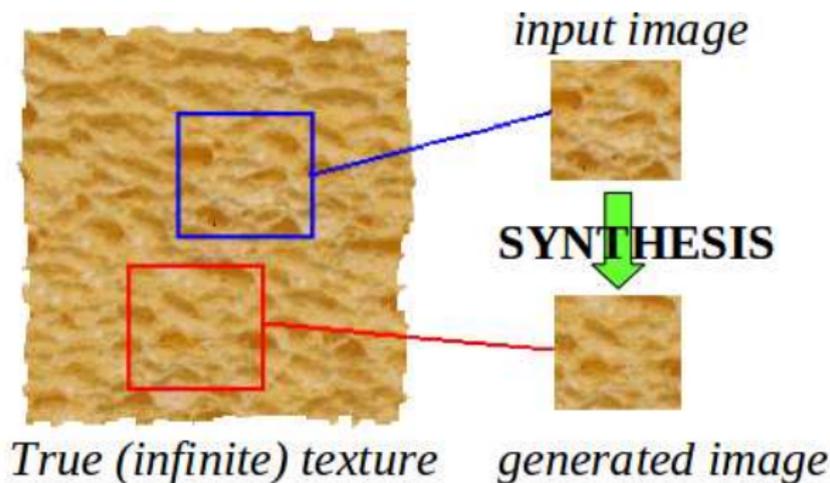
- ① Travailler ce qui est global
 - Dessiner les contours = les structures linéaires, la géométrie
 - Continuité de la structure englobant le trou, prolongement des lignes de contours
 - Régions intérieures au trou remplies par des couleurs qui matchent celles de la frontière
- ② Travailler les détails fins
 - Repérage des blocs de textures avoisinants
 - Synthèse de plus grosses régions de textures

Contents

- 1 Introduction
- 2 Synthèse de texture
- 3 Interpolation par EDP
- 4 L'union fait la force
- 5 Résultats et comparaison
- 6 Conclusion

Synthèse de texture

Qu'est-ce que la synthèse de texture ?



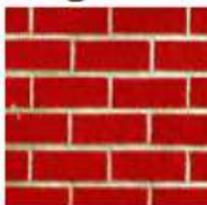
Synthèse de texture

Classification

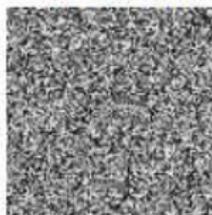
Classification

- Textures régulières : répétition de texels
- Textures stochastiques sans texels explicites
- Dans la vraie vie les textures sont un mixte de ces deux types

regular



stochastic



both?



Synthèse de texture

Interpolation de texte

“A dog is a man’s best friend. It’s a dog eat dog world out there.”

\mathbf{x}_{t-1}

a	2/3		1/3									
dog		1/3				1/3	1/3					
is	1											
man's				1								
best					1							
friend											1	
it's	1											
eat		1										
world									1			
out										1		
there											1	
.						1						
	a	dog	is	man's	best	friend	it's	eat	world	out	there	.

$p(\mathbf{x}_t | \mathbf{x}_{t-1})$

Synthèse de texture

Interpolation de texte

Chaines de Markov

- Shannon a modélisé un langage par des chaînes de Markov
- Un ensemble de mots (dictionnaire) détermine complètement la distribution de probabilité du mot suivant
- Générateur de phrases

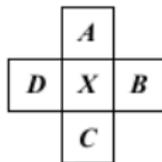
« *One morning I shot an elephant with my arms and kissed him* »

« *I spent an interesting evening recently with a grain of salt* »

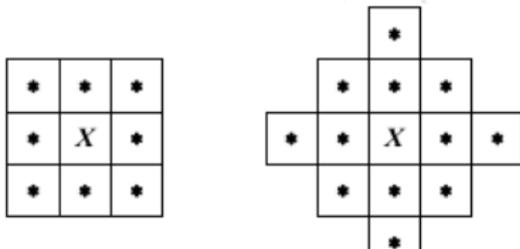
Synthèse de texture

Modélisation de texture

- Généralisation des chaînes de Markov à 2 dimensions. Probabilité $P(X|A, B, C, D)$ qu'un pixel X prenne une certaine valeur étant donnés les pixels A, B, C et D



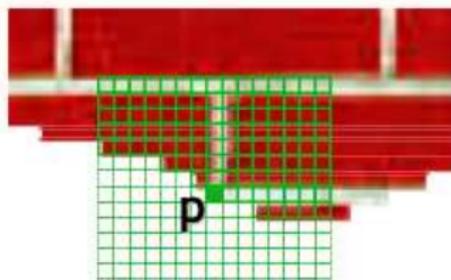
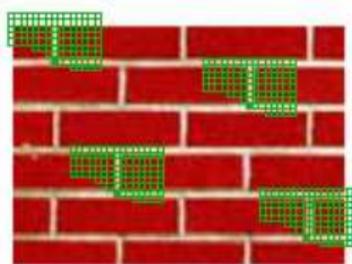
- Ordre supérieur : plus grand voisinage. $P(X|\omega(X))$



Synthèse de texture

Modélisation de texture

- Texture d'entrée $I_{smp} \subset I_{real}$.
 $\Omega(p) = \{\omega' \subset I_{real}, d(\omega(p), \omega') = 0\}$ contenant toutes les occurrences de $\omega(p)$ dans la texture infinie I_{real}
- $P(p|\omega(p))$ évaluée par histo aux centres des $\omega' \in \Omega(p)$.
- Pas accès à I_{real} ! Possible $\nexists \omega', d(\omega', \omega(p)) = 0$
- Considérer $\omega^* = \min_{\omega'} d(\omega', \omega(p)) \Rightarrow$ block matching SSD

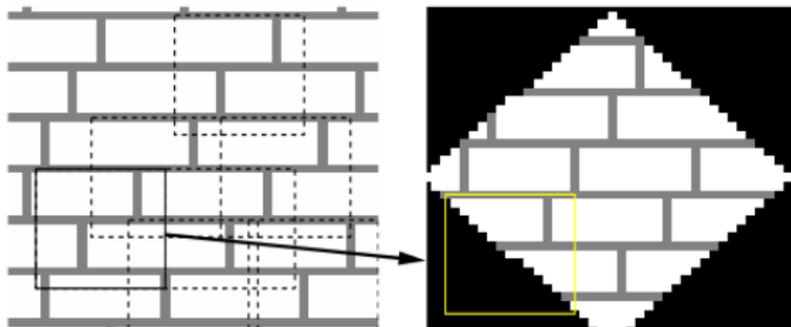


Synthèse de texture

Modélisation de texture

Algorithme d'Efros et al.

- On détermine par block matching ω^* et $\Omega'(p) = \{\omega', d(\omega', \omega(p)) < (1 + \epsilon)d(\omega^*, \omega(p))\}$
- On tire uniformément un $\omega' \in \Omega'(p) \approx \Omega(p)$ et on remplit p par le centre de ω'



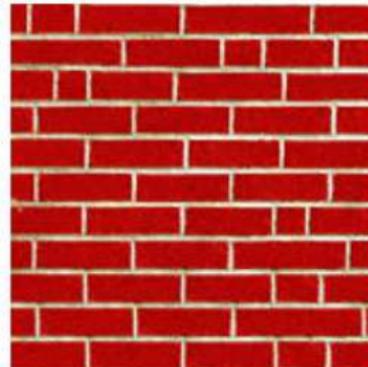
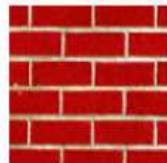
Synthèse de texture

Illustration de l'algorithme

white bread

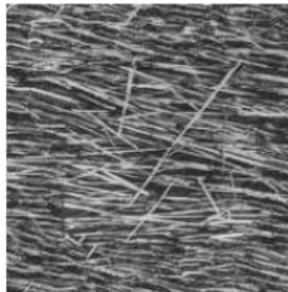
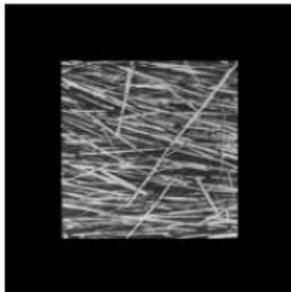
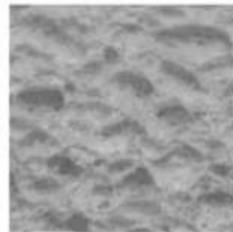
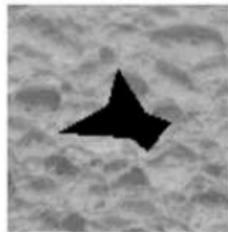


brick wall



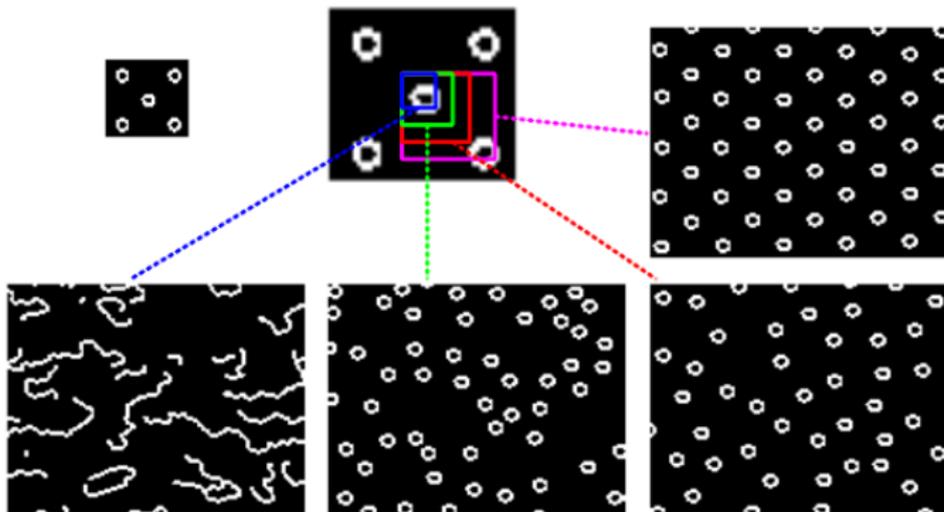
Synthèse de texture

Illustration de l'algorithme



Synthèse de texture

Illustration de l'algorithme



Synthèse de texture

Modélisation de texture

Inconvénients de l'algorithme d'Efros

- Traitement coûteux pixel par pixel
- Influence de la taille de la fenêtre
- L'ordre de traitement peut déconnecter les structures
- Ne fonctionne pas lorsque beaucoup de pixels ont été retirés

Synthèse de texture

Modélisation de texture

Inconvénients de l'algorithme d'Efros

- ~~Traitement coûteux pixel par pixel~~ \Rightarrow Algorithme d'Ashikhmin permettant la recopie par blocs
- Influence de la taille de la fenêtre
- L'ordre de traitement peut déconnecter les structures
- Ne fonctionne pas lorsque beaucoup de pixels ont été retirés

Contents

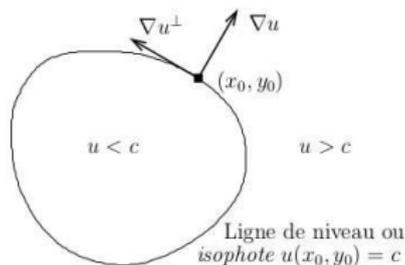
- 1 Introduction
- 2 Synthèse de texture
- 3 Interpolation par EDP**
- 4 L'union fait la force
- 5 Résultats et comparaison
- 6 Conclusion

Interpolation par EDP

Motivation

Approche variationnelle

- Le système visuel humain complète les contours occultés en utilisant des courbes d'élasticité minimisant une certaine énergie $\int |\kappa(s)|^2 ds$ où κ est la courbure.
- Prolongement des lignes isophotes (= égales intensité)
 $\frac{\partial u}{\partial t} = \nabla u^T \cdot \nabla \Delta u$ avec u connu au bord.



Interpolation par EDP

Illustration

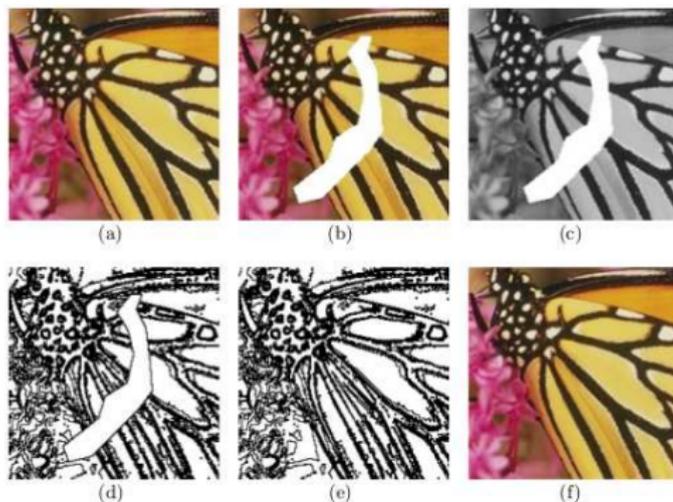


Fig. 8. [8\(a\)](#) is the original image and [8\(b\)](#) the image with occlusions in white. The luminance channel is shown in Figure [8\(c\)](#). A few isophotes are drawn in Figure [8\(d\)](#) and their reconstruction by the algorithm of Masnou and Morel [\[23\]](#) is given in Figure [8\(e\)](#). Applying the same method to the luminance, hue, and saturation channels, yields the final result of Figure [8\(f\)](#).

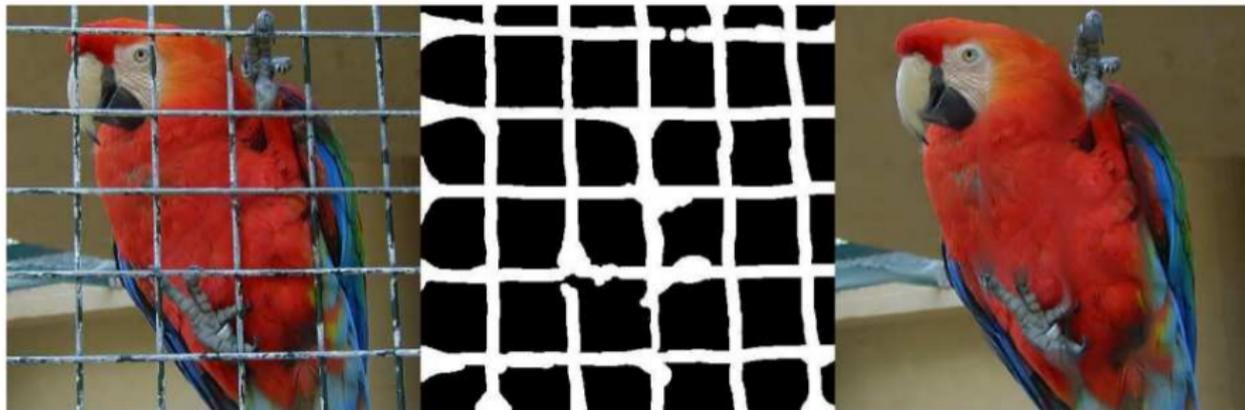
Interpolation par EDP

Avec plus de 85% de pixels supprimés



Interpolation par EDP

Problème avec les textures



Interpolation par EDP

Floutage des larges régions



Comparaison des 2 méthodes

Patch VS EDP

Avantages Patch

- Répliquer une texture
- Algorithmes par blocs

Inconvénients Patch

- Taille de la fenêtre et ordre de traitement
- Sparse data
- Ne se prête pas aux structures linéaires

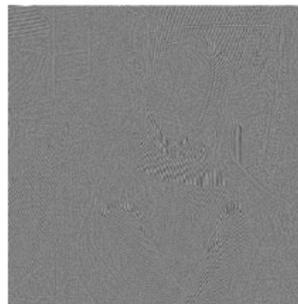
Avantages EDP

- Fonctionne très bien sur petits trous
- Cohérent avec la vue

Inconvénients EDP

- Floute les larges régions
- Résolution numérique difficiles
- Problème textures

Decomposition de l'image : géométrie + texture



Contents

- 1 Introduction
- 2 Synthèse de texture
- 3 Interpolation par EDP
- 4 L'union fait la force**
- 5 Résultats et comparaison
- 6 Conclusion

Exemplar-based inpainting

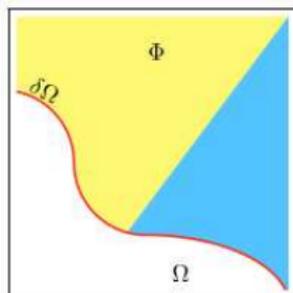
L'union fait la force

Combiner les avantages des 2 méthodes

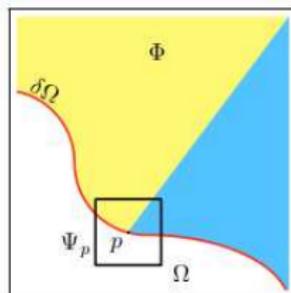
- Le procédé utilisé pour la synthèse de texture
- Rapidité de la propagation par copie de patch
- **Améliorer la gestion de l'ordre de traitement**
 - Introduire un taux de confiance en la valeur d'un pixel synthétisé
 - Propager l'information à la manière des EDP : propager les isophotes vers l'intérieur du trou

Algorithme de Criminisi

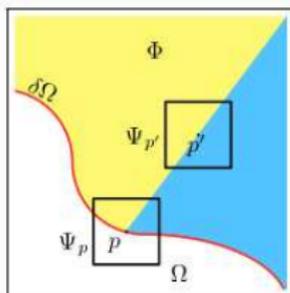
Idee générale



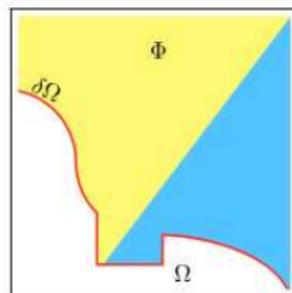
(a) Original image with *target region* Ω and its contour $\delta\Omega$ and *source region* Φ .



(b) Ψ_p marks the area we want to synthesize.



(c) The most look-alike patches for Ψ_p are located on the boundary of the two textures in the *source region* Φ , for example $\Psi_{p'}$.



(d) $\Psi_{p'}$ is copied to Ψ_p . Ω has been partly filled.

Algorithme de Criminisi

Détails

A chaque pixel à synthétiser est assigné

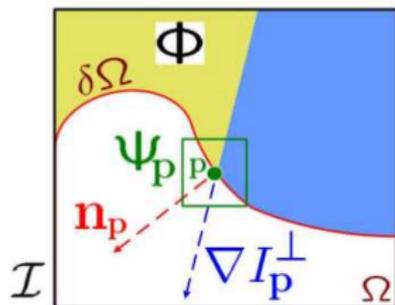
- Une valeur de couleur
- Une valeur confiance (entre 0 et 1)
- Une priorité temporaire

Etapes d'une itération de l'algorithme

- 1 Calculer les priorités de tous les pixels du front à partir des valeurs de confiance (gestion de l'ordre de traitement)
- 2 Déterminer la fenêtre prioritaire et la remplir (propagation de l'information de texture et structure)
- 3 Mise à jour des valeurs de confiance des pixels à remplir

Algorithme de Criminisi

Etape 1 : Calcul des priorités

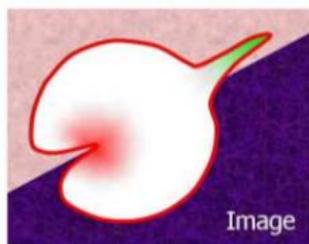
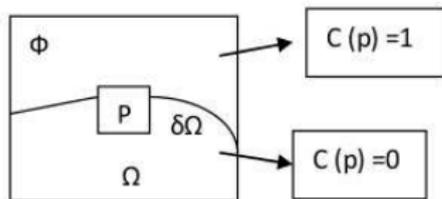


Calcul de la priorité

- Priorité $P(p) = C(p)D(p)$
- Confiance $C(p) = \frac{\sum_{q \in \psi_p \cap \Omega} C(q)}{|\psi(p)|}$
- Data $D(p) = \frac{|\nabla I_p^T \cdot n_p|}{\alpha}$

Algorithme de Criminisi

Etape 1 : Calcul des priorités



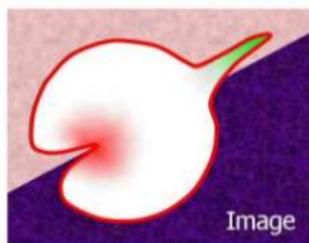
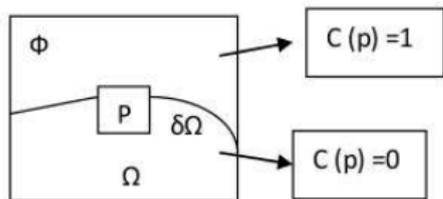
(a) Areas preferred (green) and discriminated (red) by the confidence term $C(p)$.

Calcul de la priorité

- Priorité $P(p) = C(p)D(p)$
- **Confiance** $C(p) = \frac{\sum_{q \in \psi_p \cap \Omega} C(q)}{|\psi(p)|}$
 - Taux d'information fiable entourant le pixel
 - Patch incluant coins et vaille remplis en premier
 - Renforce l'ordre concentrique (confiance décroît intérieur)
- Data $D(p) = \frac{|\nabla I_p^T \cdot n_p|}{\alpha}$

Algorithme de Criminisi

Etape 1 : Calcul des priorités



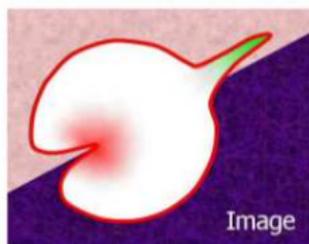
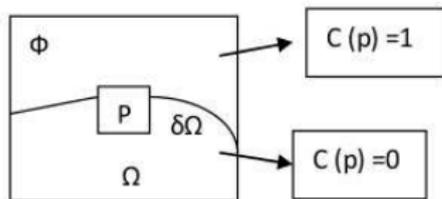
(a) Areas preferred (green) and discriminated (red) by the confidence term $C(p)$.

Calcul de la priorité

- Priorité $P(p) = C(p)D(p)$
- **Confiance** $C(p) = \frac{\sum_{q \in \psi_p \cap \Omega} C(q)}{|\psi(p)|}$
 - Taux d'information fiable entourant le pixel
 - Patch incluant coins et villed remplis en premier
 - Renforce l'ordre concentrique (confiance décroît intérieur)
- Data $D(p) = \frac{|\nabla I_p^T \cdot n_p|}{\alpha}$

Algorithme de Criminisi

Etape 1 : Calcul des priorités



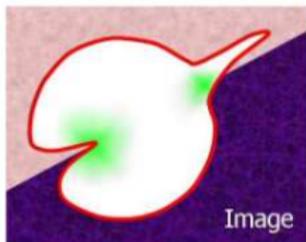
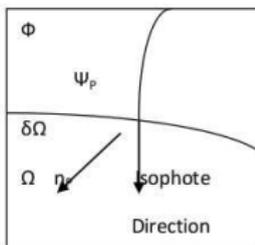
(a) Areas preferred (green) and discriminated (red) by the confidence term $C(p)$.

Calcul de la priorité

- Priorité $P(p) = C(p)D(p)$
- **Confiance** $C(p) = \frac{\sum_{q \in \psi_p \cap \Omega} C(q)}{|\psi(p)|}$
 - Taux d'information fiable entourant le pixel
 - Patch incluant coins et villed remplis en premier
 - Renforce l'ordre concentrique (confiance décroît intérieur)
- Data $D(p) = \frac{|\nabla I_p^T \cdot n_p|}{\alpha}$

Algorithme de Criminisi

Etape 1 : Calcul des priorités



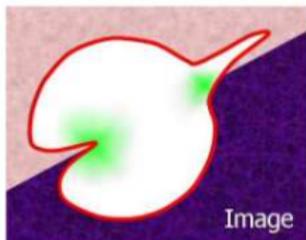
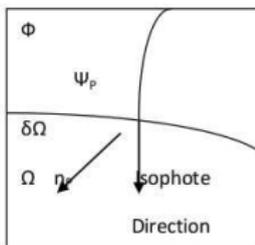
(b) Areas preferred (green) by the data term $D(p)$.

Calcul de la priorité

- Priorité $P(p) = C(p)D(p)$
- Confiance $C(p) = \frac{\sum_{q \in \psi_p \cap \Omega} C(q)}{|\psi(p)|}$
- **Data** $D(p) = \frac{|\nabla I_p^T \cdot n_p|}{\alpha}$
 - Priorise les patch incluant un flux d'isophote
 - Encourage la synthèse des structures linéaires
 - Les contours brisés tendent à se connecter

Algorithme de Criminisi

Etape 1 : Calcul des priorités



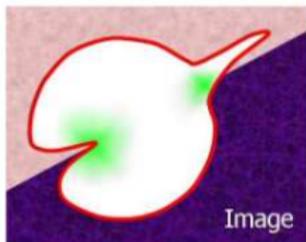
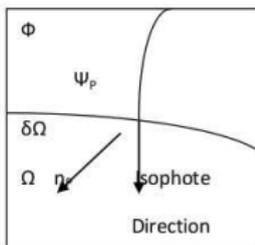
(b) Areas preferred (green) by the data term $D(p)$.

Calcul de la priorité

- Priorité $P(p) = C(p)D(p)$
- Confiance $C(p) = \frac{\sum_{q \in \psi_p \cap \Omega} C(q)}{|\psi(p)|}$
- **Data** $D(p) = \frac{|\nabla I_p^T \cdot n_p|}{\alpha}$
 - Priorise les patch incluant un flux d'isophote
 - Encourage la synthèse des structures linéaires
 - Les contours brisés tendent à se connecter

Algorithme de Criminisi

Etape 1 : Calcul des priorités



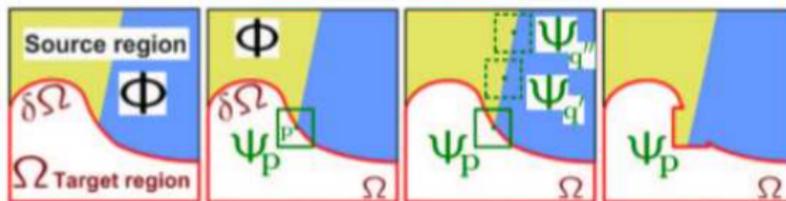
(b) Areas preferred (green) by the data term $D(p)$.

Calcul de la priorité

- Priorité $P(p) = C(p)D(p)$
- Confiance $C(p) = \frac{\sum_{q \in \psi_p \cap \Omega} C(q)}{|\psi(p)|}$
- **Data** $D(p) = \frac{|\nabla I_p^T \cdot n_p|}{\alpha}$
 - Priorise les patch incluant un flux d'isophote
 - Encourage la synthèse des structures linéaires
 - Les contours brisés tendent à se connecter

Algorithme de Criminisi

Etape 2 : propagation de l'information



- $\Psi_{\hat{q}} = \min_{\Psi_q \in \Phi} d_{SSD}(\Psi_q, \Psi_p)$
- Espace de couleur LAB utilisé pour ses propriétés de perception visuelle
- La valeur des pixels $p' \in \Psi_p \cap \Omega$ est copiée à partir des positions correspondantes dans $\Psi_{\hat{q}}$

Algorithme de Criminisi

Etape 3 : mise à jour des valeurs de confiance

Mise à jour de la confiance

Les pixels venant d'être synthétisés autour de p prennent la même confiance que p :

$$C(q) = C(p), \quad \forall q \in \Psi_p \cap \Omega$$

Contents

- 1 Introduction
- 2 Synthèse de texture
- 3 Interpolation par EDP
- 4 L'union fait la force
- 5 Résultats et comparaison**
- 6 Conclusion

Résultats et comparaison

Validation de la propagation des structures linéaires

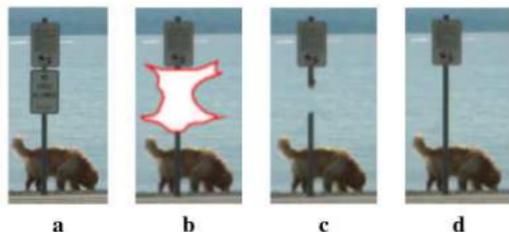


Figure 7: **Onion peel vs. structure-guided filling.** (a) Original image. (b) The target region has been selected and marked in red. (c) Results of filling by concentric layers. (d) Results of filling with our algorithm. Thanks to the *data term* in (1) the pole is reconstructed correctly.

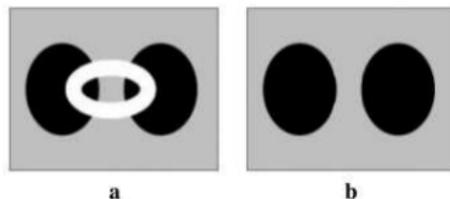


Figure 8: **Comparison with traditional structure inpainting.** (a) Original image. (b) Object removal and structure recovery via our algorithm; to be compared with fig.4 in [4].

Résultats et comparaison

Validation de la propagation des structures linéaires

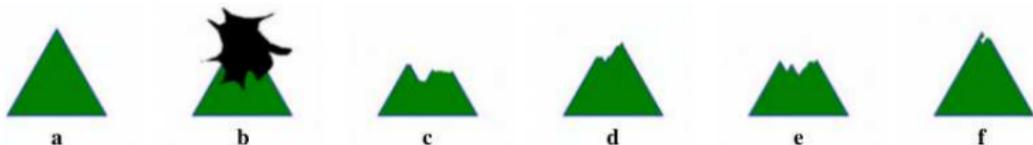


Figure 5: **Effect of filling order on a synthetic image.** (a) The original image; (b) The target region has been selected and marked in black; (c) Filling the target region in raster-scan order; (d) Filling by concentric layers; (e) The result of applying Harrison's technique which took 2' 45"; (f) Filling with our algorithm which took 5". Notice that even though the triangle upper vertex is not complete our technique performs better than the others.

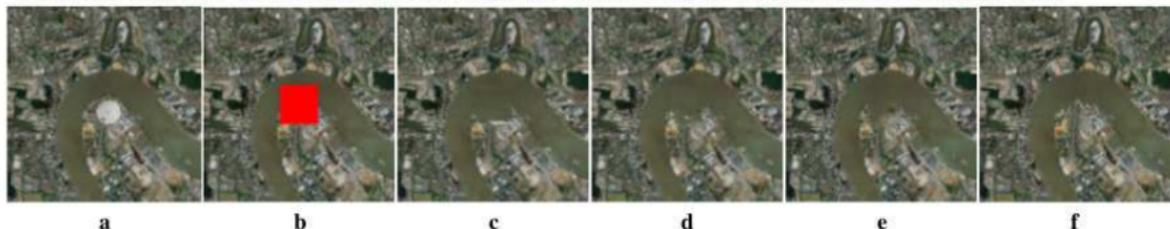


Figure 6: **Effect of filling order on an aerial photograph.** (a) The original image, an aerial view of London. (b) The target region has been selected and marked in red; Notice that it straddles two different textures; (c) Filling with raster-scan order; (d) Filling by concentric layers; (e) The result of applying Harrison's technique (performed in 45"); (f) Filling with our algorithm (performed in 2"). See text for details.

Résultats et comparaison

Evite l'effet de flou d'inpainting des larges régions

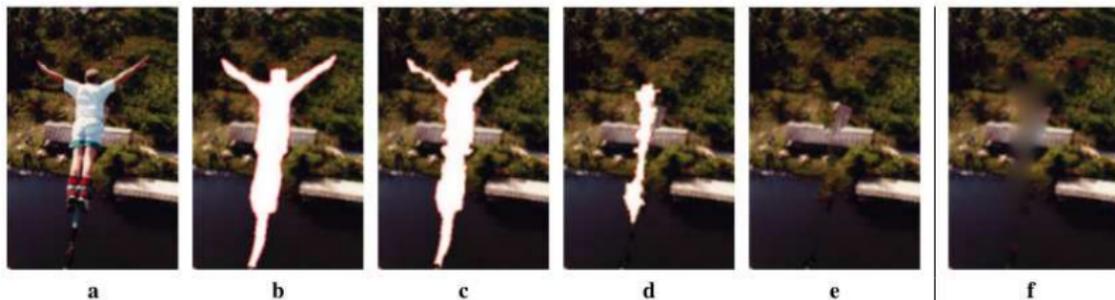


Figure 10: **Removing large objects from photographs.** (a) Original image (from [4]), $205 \times 307 \text{ pix}$. (b) The target region (in white) covers 12% of the total image area. (c,d) Different stages of the filling process. Notice how the isophotes hitting the boundary of the target region are propagated inwards while thin appendices (e.g., the arms) in the target region tend to disappear quickly. (e) The final image where the bungee jumper has been completely removed and the occluded region reconstructed by our automatic algorithm (performed in 18", to be compared with 10" of Harrison's resynthesizer). (f) The result of region filling by traditional image inpainting. Notice the blur introduced by the diffusion process and the complete lack of texture in the synthesized area.

Résultats et comparaison

Permet la suppression de texte

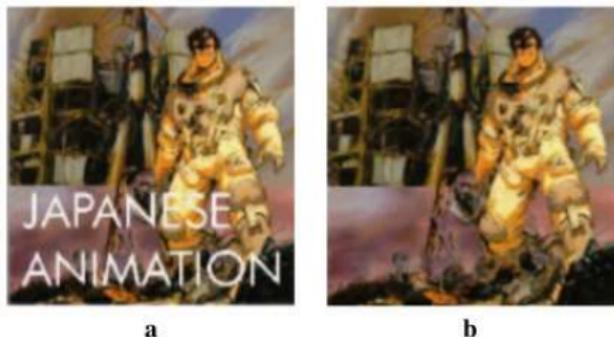


Figure 9: **Image restoration example.** (a) Original image. The text occupies 9% of the total image area. (b) Result of text removal via our algorithm.

Résultats et comparaison

Propage correctement textures ET structures linéaires

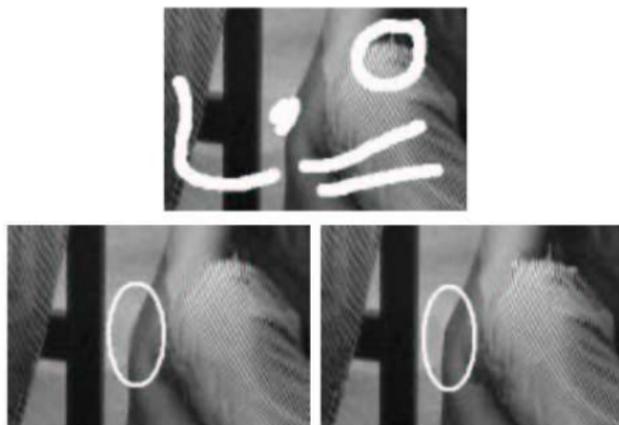


Figure 11: **Comparison with “texture and structure inpainting”.** (Top) Original image (from [5]). The target regions are marked in white. (Bottom left) Region filling via the inpainting algorithm in [5]. Notice the blur of the edge in the circled region. (Bottom right) The result of our algorithm. Both structure and texture have been nicely propagated inside the target region. The edge in the circled region is noticeably sharper.

Contents

- 1 Introduction
- 2 Synthèse de texture
- 3 Interpolation par EDP
- 4 L'union fait la force
- 5 Résultats et comparaison
- 6 Conclusion

Conclusion

Conclusion

- Présentation d'un algorithme permettant de supprimer de *larges* régions en remplaçant l'objet sélectionné par un background plausible imitant l'apparence de l'image source
- Approche par synthèse de texture modulée par un ordre de remplissage intelligible donnant priorité aux pixels de lignes isophotes
- Technique capable de propager à la fois les textures et les structures linéaires

Ouverture

Ouverture

- Propager les structures courbes correctement
- Application à l'*inpainting* de vidéos

Questions ?

