



Projet IMA : Inpainting d'images

--«Object Removal by Exemplar-Based Inpainting» by A. Criminisi,
P. Pérez, K. Toyama

Alessandro Montaldo
Chun Wu
Tuteur : Yann Gousseau

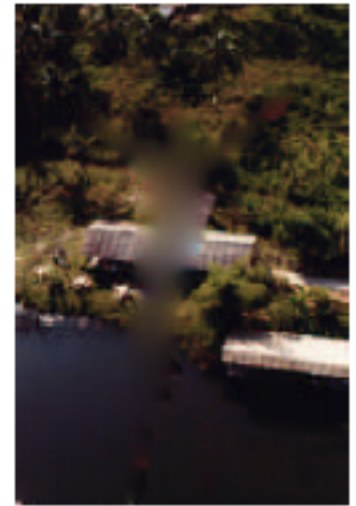
I. Introduction

La technique 'Inpainting' vise à reconstruire des zones inconnues dans une image en utilisant l'information connue de son voisinage.

État de l'art

- L'approche basée géométrie(traditional image inpainting)

➡ Flou introduit par le processus de diffusion et l'absence totale de texture dans la zone cible.



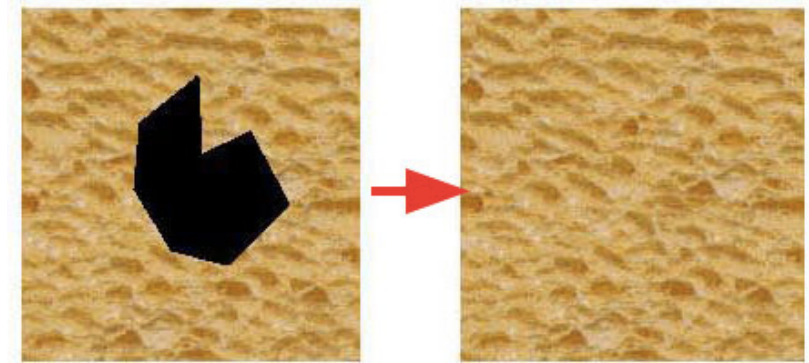
I. Introduction

État de l'art

- L'approche basée motif ou patch ("texture synthesis")

➡ Copier/coller de patches d'images en calculant la similarité des patches.

➡ Bon pour de larges zones à remplir
En revanche, ces méthodes n'assurent pas une géométrie 'globale' de la zone à remplir.



I. Introduction

En 2004, Criminisi, etc., a combiné les deux approches d'un nouvel algorithme (Exemplar-Based Inpainting) avec les avantages des deux approches.

L'essentiel de notre projet est d'implémenter l'algorithme d'Exemplar-Based Inpainting en Python.

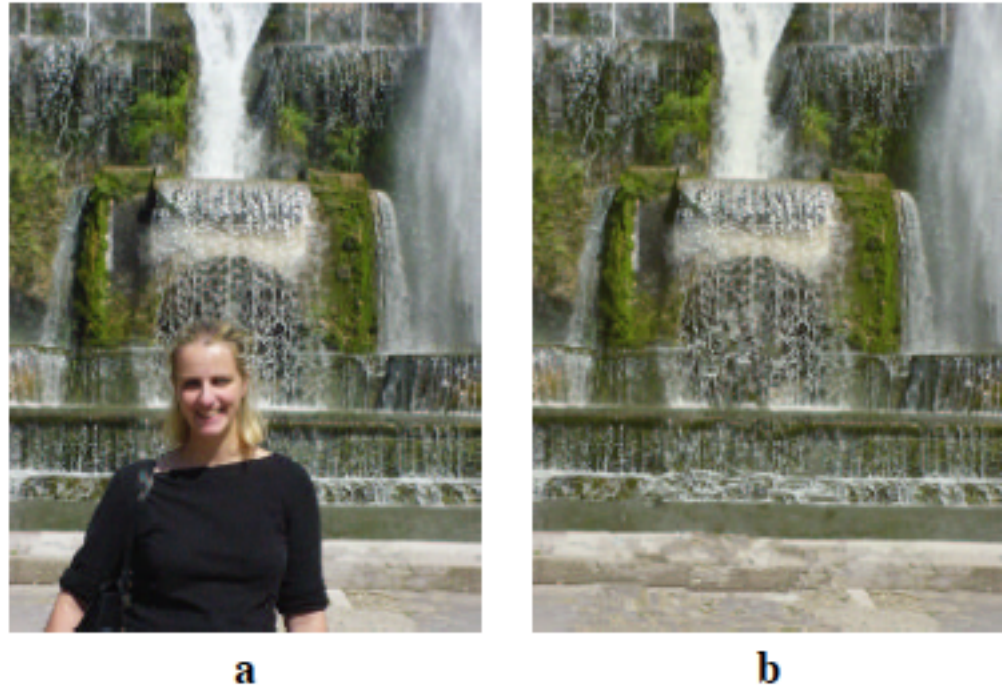
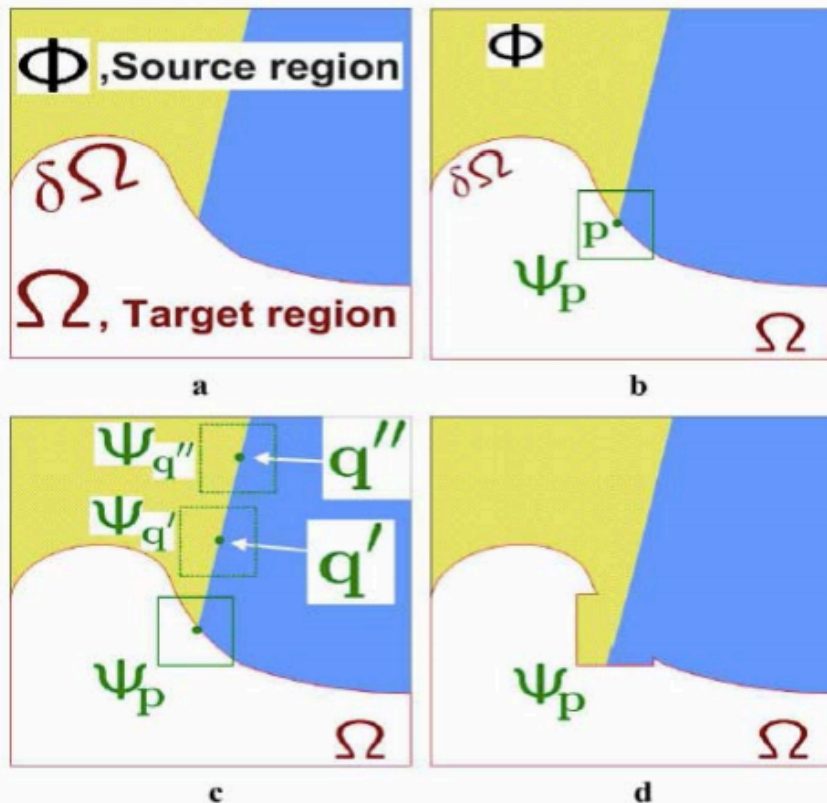


Figure 1: Supprimer des objets volumineux des images.(a) Image originale. (b) La région correspondant à la personne au premier plan a été sélectionnée manuellement, puis automatiquement supprimée.

Source : “Object Removal by Exemplar-Based Inpainting” by A. Criminisi, P. Pérez, K. Toyama

II. Algorithme de Criminisi (Exemplar-Based Inpainting)



- Extract the manually selected initial front $\delta\Omega^0$.
- Repeat until done:
 - 1a. Identify the fill front $\delta\Omega^t$. If $\Omega^t = \emptyset$, **exit**.
 - 1b. Compute priorities $P(p) \quad \forall p \in \delta\Omega^t$.
 - 2a. Find the patch $\Psi_{\hat{p}}$ with the maximum priority, i.e., $\Psi_{\hat{p}} \mid \hat{p} = \arg \max_{p \in \delta\Omega^t} P(p)$
 - 2b. Find the exemplar $\Psi_{\hat{q}} \in \Phi$ that minimizes $d(\Psi_{\hat{p}}, \Psi_{\hat{q}})$.
 - 2c. Copy image data from $\Psi_{\hat{q}}$ to $\Psi_{\hat{p}}$.
 3. Update $C(p) \quad \forall p \mid p \in \Psi_{\hat{p}} \cap \Omega$

Table 1: Region filling algorithm.

Figure 2: Propagation de la structure par synthèse de texture basée sur des exemples (patches).
– Source : “Object Removal by Exemplar-Based Inpainting” by A. Criminisi, P. Pérez, K. Toyama.

II. Algorithme de Criminisi

(Exemplar-Based Inpainting)

Détails de l'algorithme

- Calculer la priorité de patch

$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p})$$

Où $C(\mathbf{p})$ est le terme de confiance;
 $D(\mathbf{p})$ est le terme de données;
 \mathbf{p} est le pixel central d'un patch

- ✓ $C(\mathbf{p})$ -- une mesure de la quantité d'informations fiables entourant le pixel

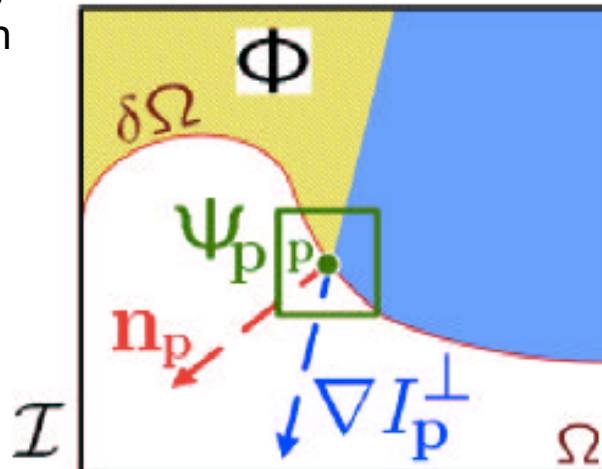
Initialement
$$C(\mathbf{p}) = \begin{cases} 0, \forall \mathbf{p} \in T \\ 1, \forall \mathbf{p} \in S \end{cases}$$

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \psi_p \cap S} C(\mathbf{q})}{|\psi_p|}$$

- ✓ $D(\mathbf{p})$ -- encourager la propagation de la structure linéaire dans la région cible.

$$D(\mathbf{p}) = \frac{|\nabla I_p^\perp \cdot \mathbf{n}_p|}{\alpha}$$

Où \mathbf{n}_p est la normale au contour $\delta\Omega$ de la région cible Ω et ∇I_p^\perp est l'isophote au point \mathbf{p} .



II. Algorithme de Criminisi

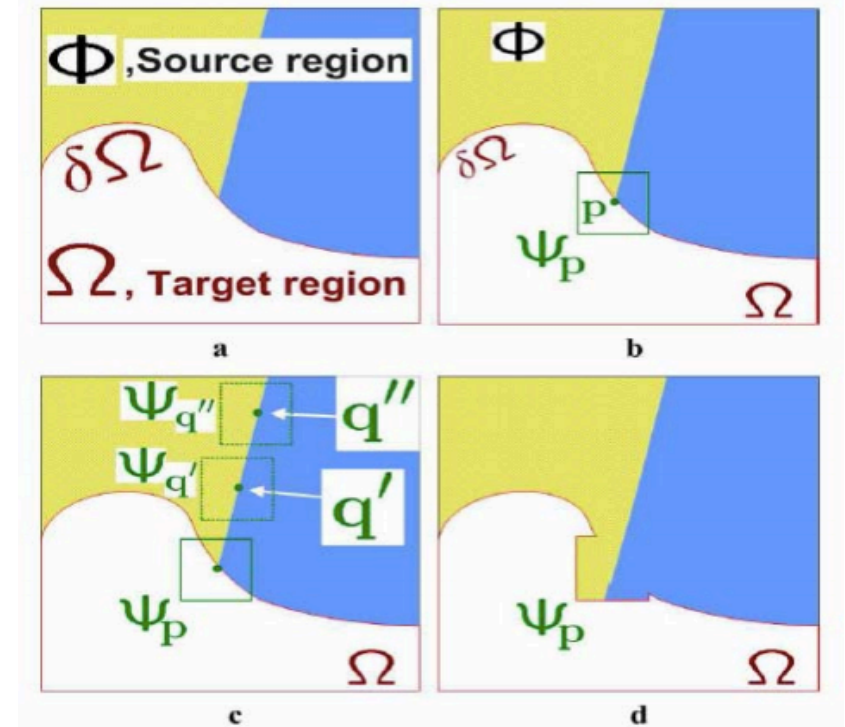
(Exemplar-Based Inpainting)

- Choisir le meilleur patch source

$$\psi_{\hat{q}} = \arg \min_{\psi_q \in \Phi} d(\psi_{\hat{p}}, \psi_q)$$

Avec $d(\psi_{\hat{p}}, \psi_q)$

la somme des différences au carré (SSD)
des pixels déjà remplis (connues) dans
les deux patches

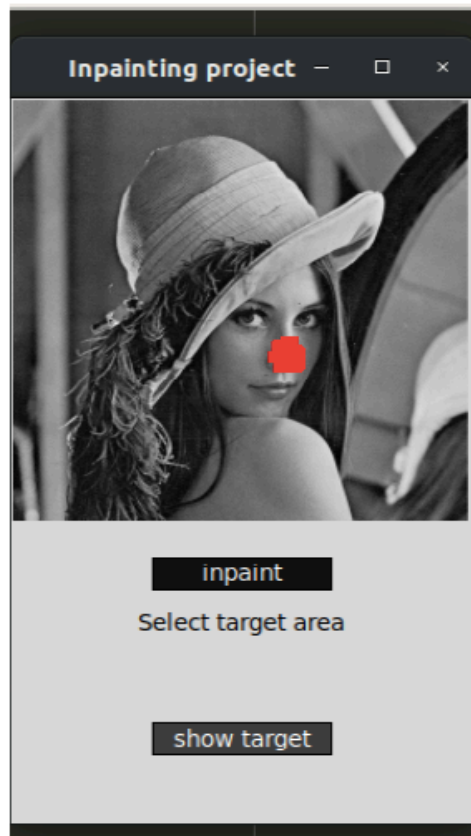


- Updating confidence values

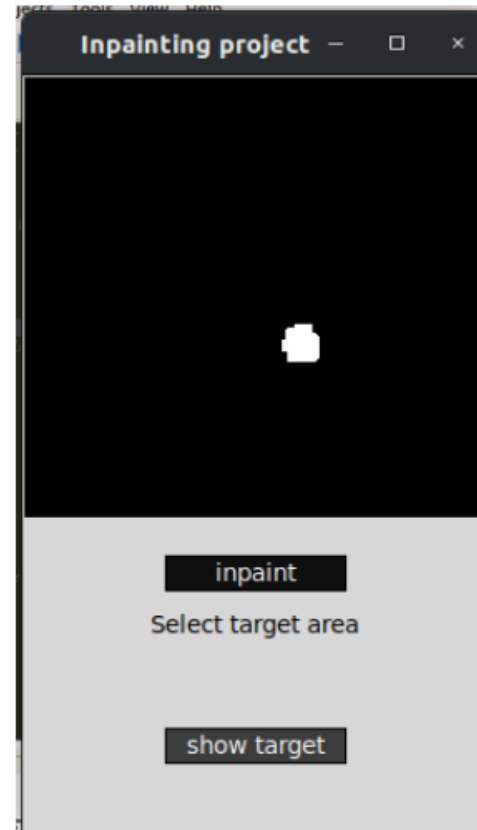
$$C(q) = C(\hat{p}) \quad \forall q \in \psi_{\hat{p}} \cup \Omega$$

III. Détails d'implémentation

➤ Interface GUI



a)



b)

Figure 3 : Captures d'écran de l'interface GUI ; a) Le région cible est dessinée en rouge à l'aide d'un curseur ; b) Le masque qu'on a dessiné est présenté en blanc.

III. Détails d'implémentation

➤ Implémentation de la fonction 'findFrontierPoints'

- Recherche horizontale
- Recherche verticale

➤ Implémentation de la fonction pour calculer la priorité de pixel

✓ $C(p)$

Mais pour $D(p)$

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

IV. Results and Discussions

➤ Effet de taille de patch

✓ Image « Église »



a)



b)



c)



d)



e)

Figure 4 : Résultats avec tailles de patch différentes ;

a) Image originale ;

b) Image avec la région cible sélectionnée en rouge ;

c) Résultats avec taille de patch de 3x3 ;

d) Patch de 5x5 ;

e) Patch 7x7

IV. Results and Discussions

➤ Effet de taille de patch

✓ Image « Chien »



a)



b)



c)



d)



e)

Figure 5 : Résultats avec tailles de patch différentes ;

a) Image originale ;

b) Image avec la région cible sélectionnée en rouge ;

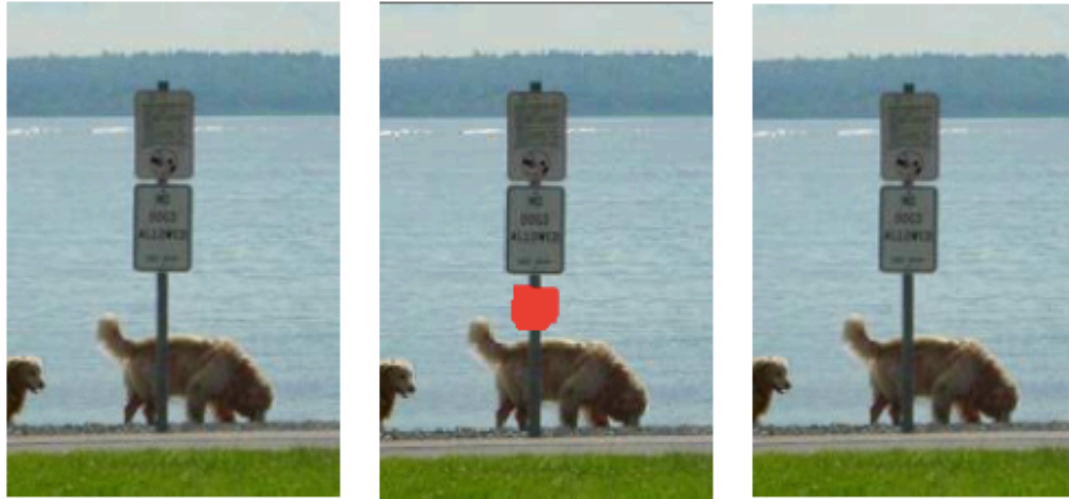
c) Résultats avec taille de patch de 3x3 ;

d) Patch de 5x5 ;

e) Patch 7x7

IV. Results and Discussions

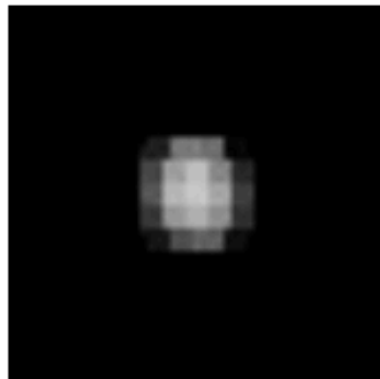
➤ Effet d'isophote



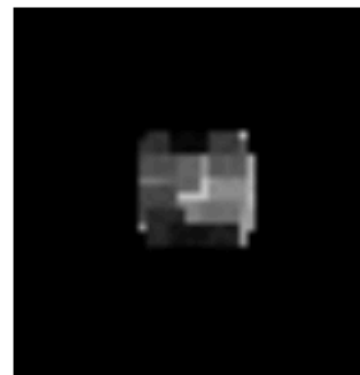
a)

b)

c)



e)



f)

Figure 6 : Effet d'isophote ;
a) Image originale ;

b) Image avec la région cible
sélectionnée en rouge ;

c) Résultat obtenu
sans/avec isophote (parall
pour les deux cas) ;

e) Ordre de remplissage (de
pixels foncés aux pixels
clairs) sans isophote ;

f) Ordre de remplissage
avec isophote

V. Conclusion

➤ Améliorations

- Terme de données $D(p)$

$$D(\mathbf{p}) = \frac{|\nabla I_p^\perp|}{\alpha} \quad \longrightarrow \quad D(\mathbf{p}) = \frac{|\nabla I_p^\perp \cdot \mathbf{n}_p|}{\alpha}$$

- Optimisation de l'implémentation de l'algorithme
- Une taille de patch adaptative



Références :

- [1] A. Criminisi, P. Perez, K. Toyama. Region filling and object removal by exemplar-based inpainting. In 2004 IEEE Transactions on Image Processing 9 1200-1212.
- [2] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In Proc. ACM Conf. Comp. Graphics (SIGGRAPH), pp. 417–424, New Orleans, LU, Jul 2000.
- [3] Jino Lee, Dong-Kyu Lee, and Rae-Hong Park. Robust exemplar-based inpainting algorithm using region segmentation. In IEEE Transactions on Consumer Electronics (Volume:58, Issue:2, May 2012), pp. 553-561, 05 July 2012.

Merci pour votre attention