

Image Quilting pour la synthèse et le transfert de texture

Membres du groupe
EL MOSSLIH Amal ROCHDI Adam

Projet Image
Traitement d'images

27 mai 2025

Sommaire

1 Introduction

2 Synthèse de texture

- Principe et algorithme
- Résultats
- Influence des paramètres
 - Taille des blocs
 - Recouvrement

3 Transfert de texture

- Principe et algorithme
- Influence des paramètres
- Résultats

4 Conclusion

5 Approches modernes

Un peu d'histoire

- Dans les années 90, Heeger et Bergen ouvrent la première voie de **la synthèse de texture** avec des méthodes basées sur des histogrammes de filtres multi-échelles.
- En 1999, Efros et Leung proposent une approche pixel-par-pixel par voisinage, innovante mais lente et peu robuste.
- En 2001, Efros et Freeman simplifient radicalement : assembler des blocs de texture comme un patchwork. C'est la naissance de l'**Image Quilting**.
→ Une méthode simple, rapide et efficace, qui prépare le terrain au transfert de texture.

Introduction

La synthèse de textures consiste à créer des images réalistes en combinant des fragments d'une texture d'origine.

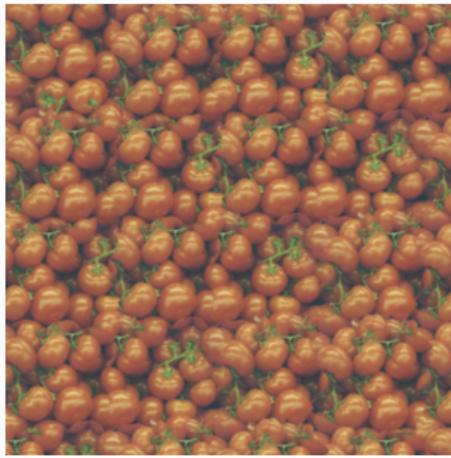


Figure 1 – Exemple de synthèse de texture.

Synthèse de texture (1/3)

Principe - Partie 1

- La synthèse débute par la découpe de l'image source en blocs carrés (*patchs*) qui se chevauchent.
- Une première méthode place ces patchs aléatoirement, ce qui donne un résultat approximatif.
- Pour améliorer, les patchs sont choisis afin de minimiser la différence dans la zone de recouvrement avec ceux déjà placés.

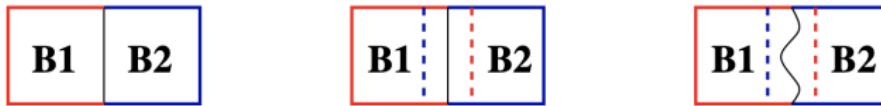


Figure 2 – Recouvrement des patchs

Synthèse de texture (2/3)

Principe - Partie 2

- La clé réside dans la **découpe minimale d'erreur** entre patchs superposés.
- Cette découpe est obtenue en calculant un chemin à coût minimal sur la surface d'erreur, via programmation dynamique :

$$E_{i,j} = e_{i,j} + \min(E_{i-1,j-1}, E_{i-1,j}, E_{i-1,j+1}).$$

- Le chemin ainsi défini délimite la frontière entre patchs.



Figure 3 – Masque binaire obtenu après la découpe minimale.

Synthèse de texture (3/3)

Algorithme de quilting de texture

- ① Pour chaque position dans l'image synthétisée
(balayage ligne par ligne, pas = taille_bloc - recouvrement)
- ② Chercher dans la texture d'entrée tous les blocs respectant les contraintes de recouvrement avec une erreur tolérée
- ③ Sélectionner au hasard un bloc parmi ceux trouvés
- ④ Calculer la surface d'erreur entre le bloc choisi et les blocs voisins
- ⑤ Trouver le chemin de coût minimal (découpe minimale d'erreur)
- ⑥ Coller le bloc découpé dans l'image synthétisée en utilisant les masques binaires.

Paramètres : Taille bloc, recouvrement 1/6, erreur L_2

Résultats(1/2)

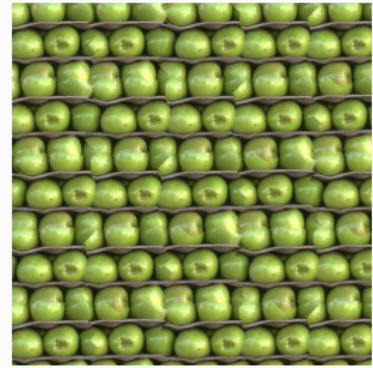


Figure 4 – Textures originales et leurs résultats synthétisés.

Résultats(2/2)



Figure 5 – Textures originales et leurs résultats synthétisés.

Influence des paramètres

La taille des blocs influence l'apparence de la texture synthétisée :

- **Petits blocs** : s'adaptent bien localement, mais peuvent fragmenter la texture ou omettre les motifs globaux si ceux-ci sont plus grands que la taille du bloc.
- **Grands blocs** : conservent mieux les structures globales, mais peuvent introduire des motifs visibles ou des transitions plus abruptes.

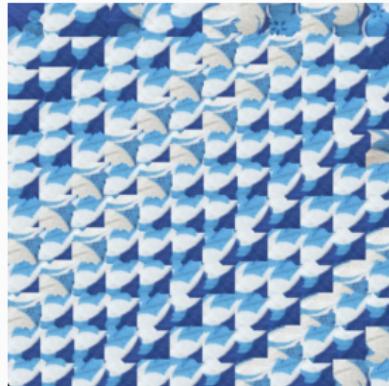


Figure 6 – Texture originale, synthèse avec blocs de taille 30x30, puis 180x180.

Influence des paramètres

Effet du recouvrement :

- **Petit recouvrement** : réduit le temps de calcul, mais peut provoquer des discontinuités visibles entre les blocs.
- **Grand recouvrement** : permet une meilleure transition entre blocs, avec des joints plus lisses, mais augmente le coût de calcul et peut introduire des répétitions.

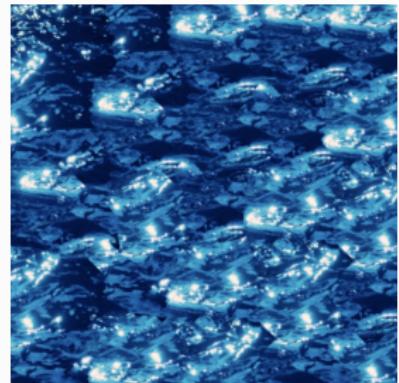
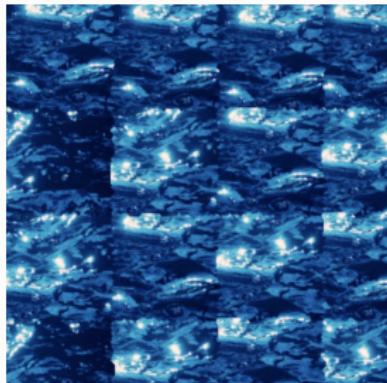
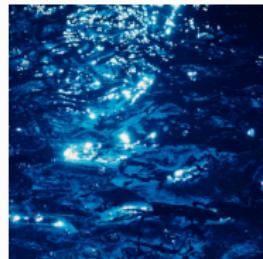


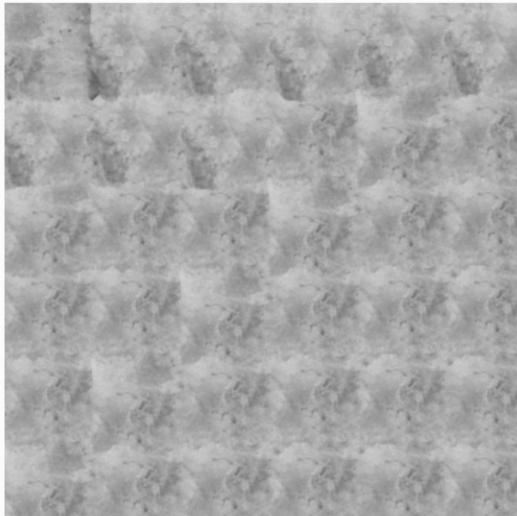
Figure 7 – Texture originale. Au centre : recouvrement = 0.05, temps = 2512 ms. À droite : recouvrement = 0.3, temps = 5913 ms.

Métriques d'évaluation

- **PSNR** : mesure l'erreur quadratique moyenne. Plus la valeur est élevée, plus les images sont similaires.
- **MSSIM** : mesure la similarité structurelle (luminance, contraste, structure). Valeur proche de 1 = haute qualité perçue.
- **evaluateSynthesisQuality** :
 - Compare l'image de référence et l'image synthétisée via une fenêtre glissante.
 - Calcule la moyenne des PSNR et MSSIM sur tous les patches.

Résultats d'évaluation

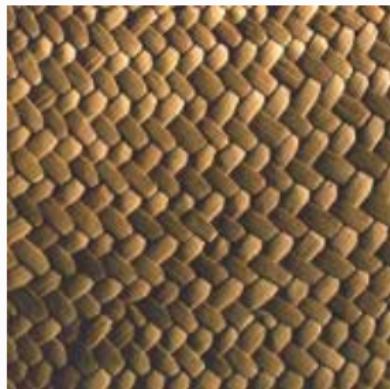
- **PSNR moyen** : 18.58 dB
- **MSSIM moyen** : (0.30, 0.30, 0.30)
- **Patches évalués** : 4514



Transfert de texture : principe

Deux contraintes à satisfaire

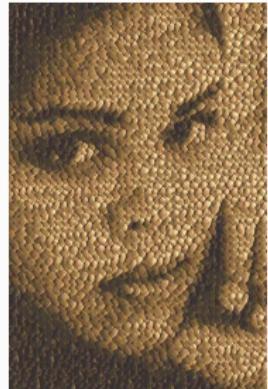
- ① Continuité locale : les blocs doivent « quilter » sans couture visible.
- ② Correspondance globale : chaque bloc doit matcher la carte de luminance (ou d'orientation) de I_c .



Texture source T_s



Image cible I_c



Résultat transféré

Étape 1 : Prétraitement

- Extraction **la luminance** de l'image cible et la texture ($(R + G + B) / 3$)
- Découpage de la texture source en patchs carrés (taille de bloc, chevauchement défini)
- Calcul de l'erreur sur le *recouvrement*
- Initialisation d'une image temporaire pour accueillir les patchs

Étape 2 : Sélection et placement des patchs

- Parcours régulier de la grille (ligne par ligne)
- **Premier patch** : celui minimisant l'erreur de luminance $E_{\text{luminance}}$
- Pour chaque position suivante :
 - Calculer l'erreur de chevauchement E_{overlap} et l'erreur de luminance $E_{\text{luminance}}$
 - Combiner les erreurs via :

$$E = \alpha E_{\text{overlap}} + (1 - \alpha) E_{\text{luminance}}$$

- Conserver les patchs avec $E \leq (1 + \text{tol}) \min E$
- Choisir un patch aléatoirement dans ce groupe

Étape 3 : Fusion et reconstitution

- Calcul d'une matrice d'erreur dans la bande de recouvrement
- Recherche du *chemin de coût minimal* (découpe de patch)
- Création d'un *masque binaire* à partir de ce chemin
- Fusion du nouveau patch : application du masque pour un joint fluide
- Recadrage final de l'image synthétisée à la taille désirée

Paramètres clés et leur rôle

- **Taille de bloc**

- Petits blocs : meilleure adaptation locale, mais perte de motifs globaux
- Gros blocs : conservation des grandes structures, mais joints plus visibles

- **Taux de recouvrement**

- Faible : calcul rapide, coutures marquées
- Élevé : transitions plus douces, coût de calcul augmenté

- **Tolérance**

- Définit l'écart maximal accepté pour constituer le pool
- Grande tolérance : plus de variations, coutures plus estompées
- Petite tolérance : sélection stricte, risque de répétitions

- **Poids α**

- $\alpha \approx 0$: priorité à la correspondance globale
- $\alpha \approx 1$: priorité à l'homogénéité des coutures

Processus itératif de transfert de texture

- ➊ Initialiser avec une grande taille de patch.
- ➋ Pour chaque itération :
 - Parcourir l'image cible.
 - Pour chaque bloc, choisir un patch source qui :
 - minimise l'erreur de recouvrement,
 - reste cohérent avec le bloc synthétisé à l'itération précédente.
 - Réduire la taille des patchs.
- ➌ Répéter jusqu'à obtenir un résultat visuellement cohérent (typiquement 3 à 5 itérations).

Résultats du transfert de texture



Texture source

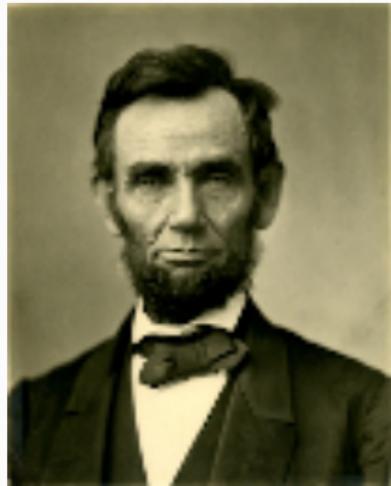


Image guide (forme)



Résultat du transfert

Résultats du transfert de texture



Texture source



Image guide (forme)



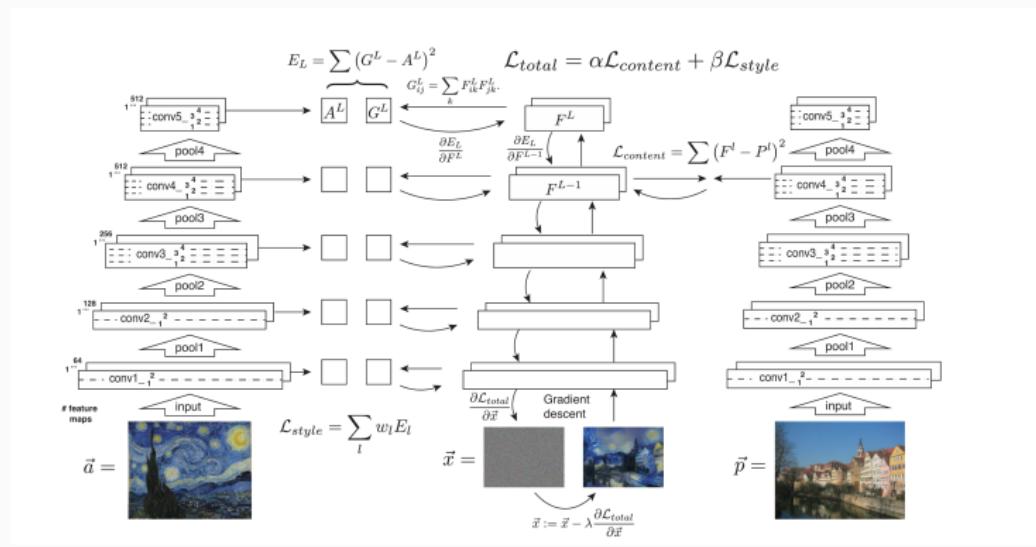
Résultat du transfert

Conclusion

- Le quilting permet une synthèse et un transfert de texture simples et visuellement efficaces.
- Les résultats sont bons pour des textures **répétitives et cohérentes**.
- La qualité dépend fortement du **choix des paramètres** : taille de patch, recouvrement, tolérance.
- Méthode efficace mais limitée pour des textures complexes.

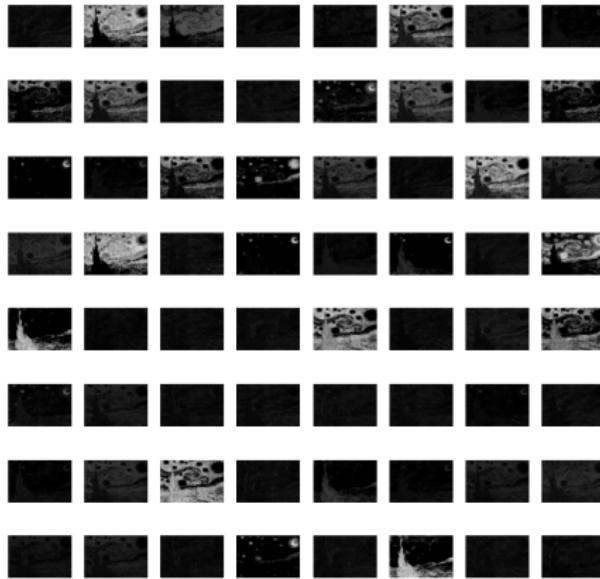
Vers des approches modernes

- En 2016, Gatys et al. introduisent une méthode révolutionnaire de **style transfer** basée sur les réseaux de neurones convolutifs.
- L'idée : séparer le *contenu* et le *style* d'une image en exploitant les représentations internes d'un CNN.



Vers des approches modernes

Features of Layer block1_conv1



Références I

-  Alexei A. Efros and William T. Freeman. *Image quilting for texture synthesis and transfer.* Proceedings of SIGGRAPH '01, ACM, 2001, pp. 341–346.
<https://people.eecs.berkeley.edu/~efros/research/quilting/quilting.pdf>
-  OpenCV. *OpenCV and similarity measurement.*
https://docs.opencv.org/4.2.0/d5/dc4/tutorial_video_input_psnr_ssim.html
-  Zhou Wang et al. *Image quality assessment : from error visibility to structural similarity.* IEEE Trans. on Image Processing, vol. 13, no. 4, 2004, pp. 600–612.
<http://dx.doi.org/10.1109/TIP.2003.819861>
-  Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric.* CVPR, 2018.
<https://github.com/richzhang/PerceptualSimilarity>
-  Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric.* CVPR, 2018.
<https://github.com/richzhang/PerceptualSimilarity>
-  Leon A. Gatys, Alexander S. Ecker, Matthias Bethge. *Image Style Transfer Using Convolutional Neural Networks.* CVPR, 2016, pp. 2414–2423.
<https://doi.org/10.1109/CVPR.2016.265>