

IM01 PROJ

Corlou Elias
Palagi Théo

October 2025

1 Détection de copier-coller

Notre projet a pour but de déterminer si une image a été modifiée, en identifiant d'éventuelles zones dupliquées. Il s'appuie principalement sur l'algorithme **PatchMatch**, qui compare des patchs de pixels (par exemple des zones de 9×9 pixels) au sein d'une même image. En regardant les déplacement entre les différents patch, nous pourrons dire si une zone a été dupliquée ou non.

2 Notre avancée

On a implémenté toutes les étapes du patchmatch, qui fonctionnent correctement. Mais pour le moment nous l'avons uniquement fait en niveau de gris. L'algorithme nous renvoie le tableau des déplacements i.e. les pixels de l'image A associés à des offsets correspondant à une position sur l'image B. Vous pouvez regarder l'avancé de notre projet sur notre dépôt GitHub suivant : github.com/y4sxk1n/IM01_PROJ.

2.1 Initialisation

Pour chaque pixel de l'image A, on attribue aléatoirement ce que l'on va appeler un offset, pointant vers une position dans l'image B, de même taille.

2.2 Propagation

En parcourant A, chaque pixel met à jour son offset en testant ceux de ses voisins déjà traités (haut/gauche, puis bas/droite) : si un offset voisin donne une meilleure correspondance, il est adopté. On mesure la similarité entre le patch centré sur ce pixel dans A et le patch correspondant dans B (SSD : somme des carrés des différences).

2.3 Recherche aléatoire

Autour du meilleur offset trouvé, on teste quelques positions aléatoires dans des fenêtres de plus en plus petites (rayon décroissant) pour échapper aux minima locaux.

2.4 Itération

On itère alors 4-5 fois les deux dernières étapes pour avoir le résultat final. Un tableau des déplacements, qui nous permettra de savoir si un des patch a été dupliqué.

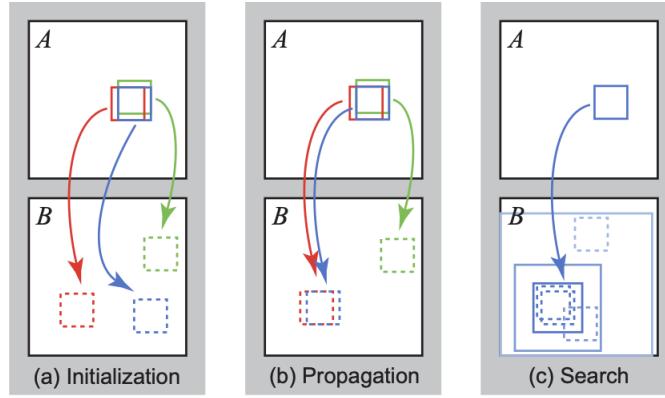


Figure 1: Les différentes étapes du patchmatch

2.5 Reconstruction

Nous avons aussi essayé de reconstruire une image à partir d'une autre image un peu différentes comme c'est le cas dans l'article. Le résultat est le suivant :



Figure 2: Image A



Figure 3: Image B

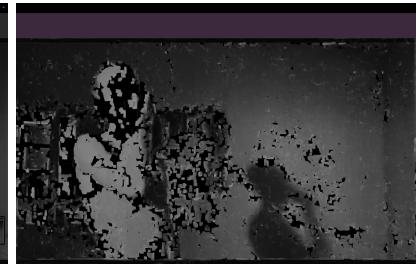


Figure 4: Image B reconstruite

On remarque que la reconstruction n'est pas encore parfaite, mais on reconnaît quand même la pose du personnage.

3 Suite du projet

- 1. Adapter l'algorithme pour les images RGB.
- 2. Tracer, comme dans l'article, les déplacements.
- 3. Optimiser le programme.
- Bonus : Faire l'implémentation GPU pour améliorer la rapidité, si le temps nous le permet.