

Coherency Sensitive Hashing

Supervision : Vinh-Thong Ta

(ta@labri.fr, <http://www.labri.fr/~ta>)

Affiliation : LaBRI, équipe Image & Son,

Pictura Research Group (<http://pictura.labri.fr>)

Mots-clés : Traitements d'images; Édition d'images; Traitements par patch; Algorithmes de correspondance; Accélération; Fonction de hachage;

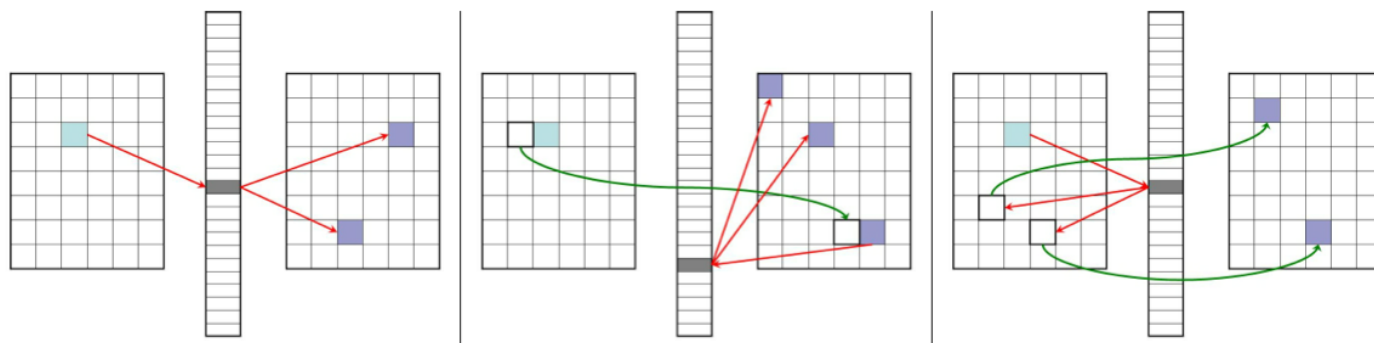


FIGURE 1 – Schéma décrivant une partie de l'algorithme proposé par [2]

Description À l'heure actuelle de nombreuses méthodes de traitements et d'édition d'image utilisent la notion de patch (petit voisinage autour du pixel à traiter) afin de caractériser de manière plus riche le pixel en lui-même (au lieu de juste utiliser la couleur ou le niveau de gris associé) dans le traitement à accomplir (filtrage, inpainting,...). Une étape du traitement est de déterminer le patch *le plus proche*, parmi tous les patches possible de l'image, du patch courant.

Calculer pour chaque pixel, son pixel le plus proche correspond à calculer une carte de correspondance appelée *NNF* (nearest neighbor field). Ce calcul induit une complexité très grande, dans la version *brute-force*, du fait que l'on va venir comparer le pixel avec l'ensemble des autres pixels (en utilisant leur patch associé).

Un des algorithmes les plus efficace et rapide de la littérature est celui proposé par [1], l'algorithme itératif du PATCH-MATCH, qui permet de calculer un *NNF* approché (et non exact) en se basant sur une notion de propagation de pixel (patch) voisin et une recherche aléatoire. Un des défauts de cette méthode est la précision du *matching* du fait de l'aléatoire introduit dans la phase de recherche. Pour palier à ce problème Korman et Avidan [2] propose de modifier la phase de recherche aléatoire par une fonction de hachage rendant la nouvelle version plus précise et permet également d'accélérer (dans les meilleurs cas) l'algorithme. La Figure 1 illustre une partie de l'algorithme proposée.

Objectifs Les objectifs de ce projet sont

- d'étudier l'article de Korman et Avidan [2] et de comprendre l'algorithme proposé;
- les auteurs proposent une implémentation de la méthode (www.eng.tau.ac.il/~simonk/CSH/) Le but est d'étudier et de comprendre le code source fourni, de tester et de confirmer les résultats obtenus par les auteurs. Enfin on proposera une implémentation simplifiée de la méthode permettant de calculer un *NNF* à l'aide de l'algorithme. Le choix du langage de programmation sera à déterminer entre l'encadrant et le groupe d'étudiants.

Références

- [1] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. PatchMatch : A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), August 2009.
- [2] Simon Korman and Shai Avidan. Coherency sensitive hashing. In *Proceedings of the 2011 International Conference on Computer Vision, ICCV '11*, pages 1607–1614, 2011.