

Compte Rendu Semaine 1 - Projet IMAGE

HAI809I - Master 1 Imagine - Université de Montpellier

Laporte Laëtitia - Chaudillon Luca

3 Mars 2024

Résumé

L'objectif de ce projet est de compresser des images réelles (issues d'APN) à partir d'une approche de superpixels.

Table des matières

1	Introduction	2
1.1	Que sont les superpixels ?	2
1.2	Informations supplémentaires	2
2	Motivations	2
3	Etat de l'art	2
3.1	Simple Linear Iterative Clustering (SLIC)	2
3.2	Variantes de SLIC	3
3.3	Autres pistes	3
4	Prévisions	4
5	Références	4

1 Introduction

1.1 Que sont les superpixels ?

Les superpixels correspondent à de petits groupes homogènes de pixels voisins. Une fois assemblés, ils permettent de retrouver les différents objets constituant une image. Ils sont principalement utilisés pour des raisons d'efficacité. Comme chaque superpixel regroupe quelques centaines de pixels similaires, manipuler les superpixels à la place des pixels réduit considérablement le nombre d'entités à traiter.

1.2 Informations supplémentaires

Les codes produits lors de ce projet ainsi que les comptes rendus et la base de données d'images que nous allons utiliser sont retrouvables sur un répertoire github à l'adresse suivante : <https://github.com/cygne1110/projet-IMAGE-M1>

2 Motivations

Nous avons choisi ce sujet, car nous avons trouvé intéressant de pouvoir comparer l'utilisation de méthode de compression (voir même de méthode d'analyse) sur une image segmentée en superpixels et une image neutre, notamment pour l'impact sur les performances ainsi que sur la qualité des résultats.

Nous avons aussi trouvé intéressant de comparer les méthodes de segmentation en superpixels entre elles (surtout, qu'après recherche, il y beaucoup de méthodes variées qui existent) pour voir quelle était la meilleure, et pour aller plus loin, voir s'il existe des méthodes qui étaient plus adaptées à certaines méthodes de compression, et pouvoir choisir la méthode de segmentation en superpixels selon la méthode de compression.

3 Etat de l'art

Nous avons recherché les différentes méthodes de segmentation en superpixels existantes, et 4 d'entre elles sont revenues le plus souvent. Nous n'avons malheureusement pas eu le temps de les étudier en détail pour ce premier compte rendu, mais nous pouvons au moins détailler celle qui nous paraît être la plus simple : la méthode SLIC.

3.1 Simple Linear Iterative Clustering (SLIC)

L'algorithme SLIC est basé sur de la classification K -mean, où chaque pixel est rattaché à une classe de pixel. Cependant, au lieu de rattaché n'importe quel pixel avec n'importe quelle classe (comme pour la compression palette), on ne compare les pixels qu'avec les classes ou les clusters avoisinants.

SLIC prend en entrée plusieurs paramètres : N le nombre de pixels de l'image et k le nombre de superpixels souhaité. On aura donc en sortie k superpixels de taille similaire. Il est à noter que l'image en entrée doit être préalablement convertie dans l'espace couleur CIELAB, un espace chromatique qui répartit les couleurs de manière plus conforme à la perception des écarts des couleurs du système visuel humain. L'espace CIELAB permet donc d'avoir des mesures de distances entre deux couleurs plus précises que l'espace RGB.

L'algorithme consiste à classifier les pixels en utilisant un k -mean, en commençant par choisir des clusters carrés de taille $S * S$ (avec S la taille moyenne des k superpixels pour qu'ils soient tous de même taille dans une image de N pixels, $S = \sqrt{\frac{N}{k}}$) et d'assigner le centre de ces clusters à la position minimale du gradient local de ce cluster (pour éviter d'avoir des centres sur le bord de l'image).

Ensuite, pour chaque itération, pour chaque pixel, on classe le pixel courant au centre de cluster le plus proche, en prenant la distance minimale avec les centres de clusters voisins (dans une région autour du pixel courant de $2S * 2S$, pour ne pas parcourir toute l'image et réduire la complexité). La distance entre deux pixels est calculée en mesurant d'abord l'écart spatial (position du pixel dans l'image) et l'écart de couleur, que l'on combinera ensuite en les normalisant selon la taille maximale de chaque écart (S pour l'écart spatial, et une constante définie par l'utilisateur pour l'écart de couleur). On répète cette opération jusqu'à convergence. Il est à noter que pour SLIC, il est nécessaire de faire une passe post traitement pour traiter certains pixels qui ne sont pas connectés aux superpixels.

3.2 Variantes de SLIC

- SLICO (Zero parameter SLIC), variante de l'algorithme SLIC qui adapte la compacité de chaque superpixel, mais qui garde une taille de cluster égale sur tous les superpixels. SLICO change notamment les méthodes de calculs de distance entre deux pixels, en calculant adaptivement l'écart de couleur maximal entre deux pixels selon les itérations précédentes.
- ASLIC (Adaptative-SLIC), adapte la compacité et la taille de chaque superpixel. ASLIC calcule adaptivement l'écart de couleur maximal et l'écart spatial maximal entre deux pixels selon les itérations précédentes.
- GSLIC (Geodesic-SLIC), adapte la compacité et la taille de pas de chaque superpixel, comme ASLIC, sauf que cette fois la distance entre deux pixels est calculé avec la distance géodésique entre deux couleurs, ce qui permet d'ignorer la distance spatiale. Cette méthode permet d'éviter d'avoir une passe de post-traitement contre un coût de calcul plus conséquent.

3.3 Autres pistes

Les 3 autres méthodes que nous avons découvertes durant nos recherches mais que nous n'avons pas pu étudier en détail sont les suivantes :

- Algorithme de Felzenszwalb (Segmentation efficace avec un graphe)
- Segmentation Quick Shift (Segmentation basée sur les k-means)
- Segmentation par ligne de partage des eaux (Watershed)

On peut déjà avoir une idée des résultats des différentes méthodes d'après la figure 1.

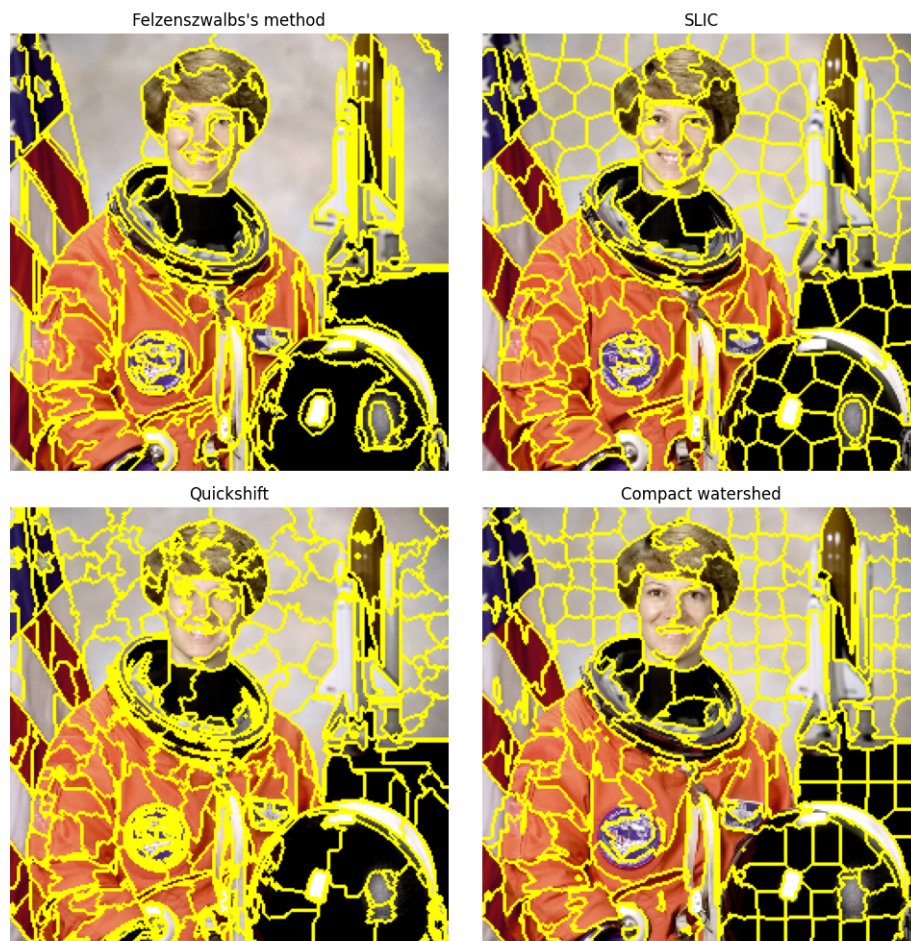


FIGURE 1 – Résultats des différentes méthodes recherchées sur une même image ([Source](#))

4 Prévisions

Pour la semaine prochaine, nous aimerions commencer à implémenter du code (notamment l'algorithme SLIC, à minima la conversion RGB vers CIELAB nécessaire à l'algorithme). Nous aimerions aussi avancer sur les études de méthodes de segmentation en superpixels.

5 Références

1. <https://makina-corpus.com/data-science/localisation-dun-objet-par-classification-de-superpixels>
2. Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Suesstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, TPAMI, May 2012.
3. <https://www.epfl.ch/labs/ivrl/research/slic-superpixels/>
4. CIELAB Color Space
5. Efficient graph-based image segmentation, Felzenszwalb, P.F. and Huttenlocher, D.P. International Journal of Computer Vision, 2004

6. Quick shift and kernel methods for mode seeking, Vedaldi, A. and Soatto, S. European Conference on Computer Vision, 2008
7. Peer Neubert & Peter Protzel (2014). Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms. ICPR 2014, pp 996-1001.