

HAI809I — Projet Image et Compression

Parcours Imagine 2024-2025

Mathis Duban (mathis.duban@etu.umontpellier.fr)
Paul Deligne (paul.deligne@etu.umontpellier.fr)

April 6, 2025

Compression basée super-pixels

Compte rendu du 6 avril 2025

1 Travail déposé dans le dépôt GitHub

Durant cette semaine de travail, de nouveaux éléments ont été déposés dans différents dossiers :

- Le dossier **Image** contient les images d'entrées et de sorties utilisées pour l'algorithme.
- Le dossier **Code** contient tout le code (fichiers .cpp/.h)

Voici l'adresse de notre repository GitHub contenant tous les éléments actuels du projet:

<https://github.com/Akkuun/Super-Pixel-Project>

2 Travail effectué cette semaine

Le travail de cette semaine a été dédié à implémenter une méthode de compression de l'image modifiée avec notre méthode de Super pixel. Nous avons également commencé l'implémentation d'une interface pour notre projet.

2.1 Travail effectué auparavant

A la fin de la semaine dernière, nous avions terminé d'implémenter deux méthodes de transformation de l'image basées sur les super pixels, les algorithmes SLIC et Meanshift. Pour la partie compression du projet, nous avions une première méthode où l'on réduisait le nombre de bits sur lesquels était codée la chrominance.

2.2 Compression par palette de l'image Super Pixel

Comme dit plus haut, nous avions implémenté une méthode de compression pour nos images super pixel. Cependant, voulions implémenter une méthode de compression vue en cours et nous avons choisi la compression par palette qui semblait être la plus adaptée à notre projet. Dans cette partie, nous allons discuter de la méthode de compression que nous avons choisie pour l'image après un des deux algorithmes et l'implémentation de cette méthode.

Notre méthode de compression choisie est la compression par palette, elle fonctionne de la manière suivante : on initialise 256 clusters et leurs centroïdes pour stocker 256 couleurs différentes de l'image super pixel. Ensuite, on associe chaque pixel de l'image à son centroïde le plus proche. Puis on met à jour les centroïdes et on recommence jusqu'à ce que l'on ait une convergence.

Voici le résultat que l'on obtient après la compression et décompression sur l'image LezardHD (3456×5184) avec $k=1000$ et $m=10$: Le PSNR entre l'image d'origine et l'image après SLIC est de 26.2 dB et le PSNR entre l'image d'origine et l'image compressée est de 25.6 dB. Si on applique l'algorithme d'Huffman sur l'image compressée, on obtient un taux de compression égal à 1,51 contre 1,13 sans.



Figure 1: LezardHD original



Figure 2: LezardHD après SLICC



Figure 3: LezardHD après SLICC et la compression palette



Figure 4: Carte des couleurs de LezardHD SLIC compressée

2.3 Interface

En plus de la compression par palette, nous avons commencé l'implémentation d'une interface pour notre projet. Voici une capture d'écran de notre interface.

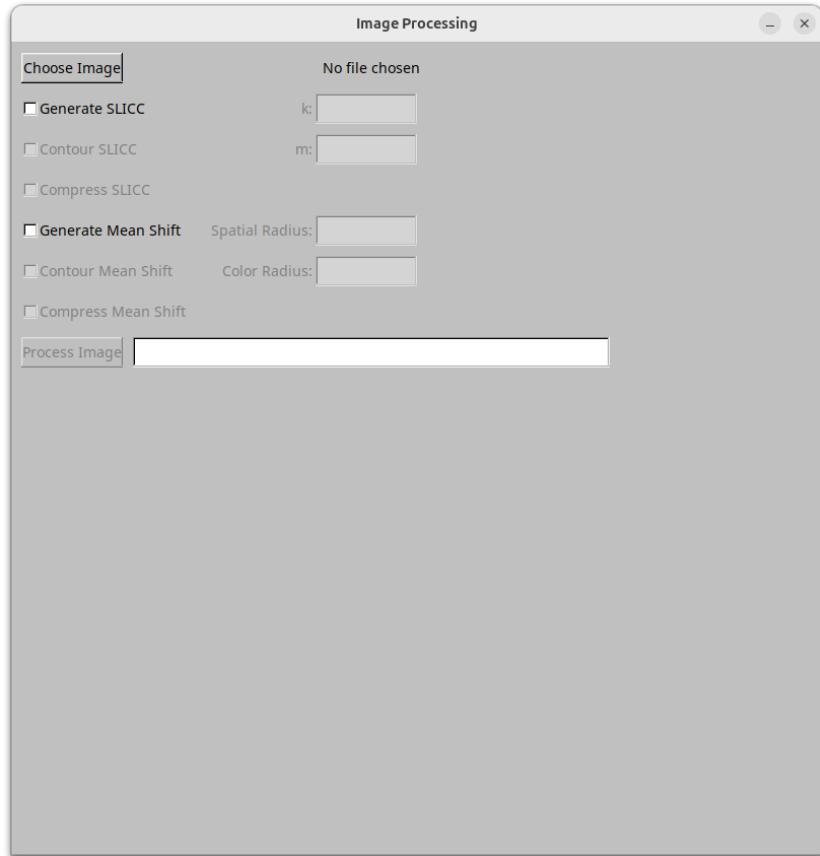


Figure 5: Capture d'écran de l'interface

L'utilisateur peut choisir une image au format PPM dans ses fichiers en cliquant sur le bouton en haut à gauche "Choose Image". Ensuite, il choisit s'il génère une image à partir de SLIC et/ou de Mean Shift, il peut ensuite définir les paramètres des différentes méthodes et s'il génère une image des contours ainsi que s'il applique la compression par palette. Enfin, une fois que tout est prêt, il peut appuyer sur le bouton "Process Image" qui va lancer le programme.

3 Travail à venir

Pour le travail à venir, nous allons terminer l'interface et vérifier qu'il n'y ait pas de bug dans notre code. Enfin, nous finirons de préparer notre présentation finale.

4 Références

References

- [1] Bibilothèque d'image Unsplash,
<https://unsplash.com/fr/s/photos/reptile>
- [2] jflalonde,
<http://vision.gel.ulaval.ca/~jflalonde/cours/4105/h17/tps/results/projet/111063028/index.html> *Segmentation d'images en superpixels via SLIC*
- [3] jflalonde, epfl
<https://www.epfl.ch/labs/ivrl/research/slic-superpixels/#SLICO> *SLIC Superpixels*