

# HAI809I — Projet Image et Compression

## Parcours Imagine 2024-2025

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

March 15, 2025

## Compression basée super-pixels

Compte rendu du 2 mars 2025

# 1 Optimisations du temps de calcul pour SLICC

La semaine précédente, nous avions terminé notre première méthode de transformation en super-pixel et nous l'avions testée sur des petites images. Cependant, nous avions constaté des durées d'exécution importantes pour l'exécution de SLICC même sur des petites images, par exemple pour des 500x500 on mettait à peu près 3 minutes d'exécution.

Le travail de cette semaine a été d'optimiser les performances du programme afin de réduire le temps de calcul. Pour cela, nous avons implémenté les éléments suivants :

- `#pragma omp parallel for` pour paralléliser les boucles et répartir les itérations entre plusieurs threads afin d'améliorer les performances.
- `#pragma omp parallel for reduction(+:deltaCk)` pour effectuer une réduction parallèle sur la variable `deltaCk`, en s'assurant que les contributions des différents threads sont correctement accumulées sans conflits d'accès mémoire.
- `#pragma omp parallel for collapse(2)` pour paralléliser efficacement des boucles imbriquées, en combinant les deux premières boucles pour mieux équilibrer la charge entre les threads.
- suppression des appels de calculs répétitifs et optimisation des appels afin d'éviter des calculs supplémentaires tout en réduisant l'empreinte mémoire ainsi que la latence d'accès aux données.
- ajout d'options de compilation dans le `Makefile` afin d'indiquer au compilateur d'optimiser les appels de boucles, d'optimiser les instructions processeur, etc. (`-O3 -floop-funroll-loops -fopenmp`), ce qui permet d'améliorer les performances globales du programme.

Après avoir ajouté ces spécifications dans le code/makefile, nous avons observé des temps de calcul significativement plus bas. Par exemple, pour une image 500x500 nous sommes passés de 3 minutes à environ 20 secondes (c'est plus d'un facteur 9).

## 2 Fonctionnement de SLICC pour des images 4K

Un de nos objectifs principaux a été d'appliquer la transformation en super-pixel sur des images 4K qui sont assez larges. Lorsque nous lancions notre programme sur des images larges, notre programme plantait. Nous avons corrigé notre programme dans lequel certains indices étaient mal accédés. Voici ce que nous obtenons maintenant pour des images 4k (15 minutes de temps de calculs) :

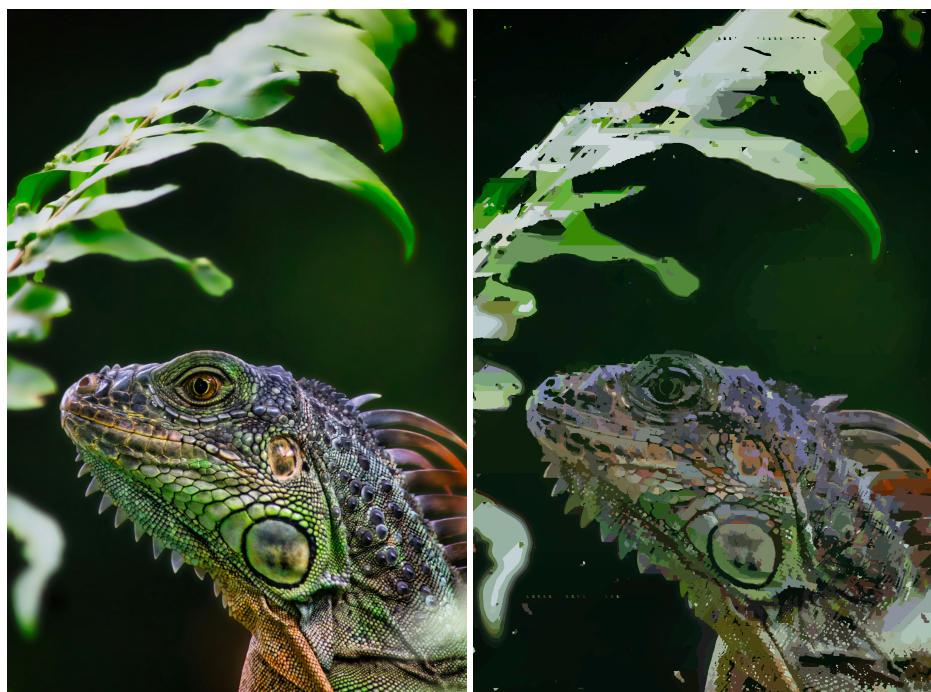


Figure 1: Image 4k avant l'application de SLICC de taille 5152x3150  
Figure 2: Image 4k après l'application de SLICC avec k=10000 et m=20

### **3 Travail à venir**

Pour le travail à venir, voici ce que nous avons de prévu :

- tester une première méthode de compression et mesurer le PSNR que nous obtenons
- commencer à développer une seconde méthode de transformation en super-pixel

## 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*