

# Super-Pixels Compte Rendu 6

Souvignet Nathan, Vaillant Hugo

Avril 2025

## Contents

<b>1</b>	<b>Implémentation de la partie statistique</b>	<b>2</b>
1.1	Sélection des images . . . . .	2
1.2	Modification de notre programme principal . . . . .	3
1.3	Automatisation . . . . .	3
1.4	Résultat . . . . .	4
<b>2</b>	<b>Interface graphique</b>	<b>6</b>

# 1 Implémentation de la partie statistique

Afin de comparer les algorithmes, le programme a été lancé plusieurs fois avec différents paramètres et différentes images. En raison du de tests, L'utilisation de scripts bash a permis de lancer en parallèle un grand nombre de tests.

## 1.1 Sélection des images

Les images ont été sélectionnées à partir du set de donnée BSDS500[1], qui contient des images naturelles et leurs contours tracés par des humains. Cela permet de mesurer la métrique de performance **Boundary Recall**. De plus, cela nous a permis d'avoir la même taille d'image pour l'entièreté de nos tests.



Figure 1: Images naturelles tirées de la banque d'images



Figure 2: Images des contours réalisés par des humains

## 1.2 Modification de notre programme principal

Le programme principal a été modifié pour qu'il effectue une série d'étapes à chaque fois qu'il sera appelé pour une image donnée :

- Récupère l'image en entrée, le nombre de superpixels, ainsi que le facteur d'adhésion au contour;
- Pour chaque algorithme (SNIC, SLIC, WaterPixel) :
  - En entrée, ils récupèrent l'image d'entrée, le nombre de superpixels, le facteur d'adhésion au contour, ainsi que le chemin de sortie pour l'image compressée.
  - Le programme renvoie la taille de l'image compressée, le PSNR (en dB) calculé entre l'image d'entrée et l'image de sortie de l'algorithme, ainsi que sa métrique de Boundary Recall.
- Une fois ces 3 valeurs récupérées pour chaque algorithme, on écrit dans le fichier de sortie (.data) nos résultats.

Exemple de données récupérées pour une image :

```
SNIC :
  k :100 m :0.5 psnr :19.4011 SizeCompressed : 51698 Recall :0.581761
SLIC :
  k :100 m :0.5 psnr :27.4527 SizeCompressed : 107471 Recall :0.93863
WATERPIXEL :
  k :100 m :0.5 psnr :18.9578 SizeCompressed : 49191 Recall :0.413752
```

## 1.3 Automatisation

A l'exécution du programme principal, un fichier contenant les données est créé pour chaque image. Les scripts bash utilisés pour des tests de PSNR, sont réutilisés et adaptés.

Le script qui permet de lancer les exécutions ('scriptMakeStatImages.sh'), écrit tout d'abord toutes les commandes dans un fichier, qui est ensuite passé à GNU Parallel. Cela permet de paralléliser un grand nombre de tests avec différentes images et différents paramètres.

Une fois les images superpixels, et le fichier '.data' contenant toutes les données créés, l'exécution du script ('scriptGetAllData.sh') permet de faire un parcours récursif dans tous les dossiers et sous-dossiers, pour récupérer les informations contenues dans les fichiers '.data' et les stocker dans un fichier unique : *all\_data\_values.data*.

Voici un extrait des lignes qu'on peut trouver dans *all\_data\_values.data* :

```
./output_stat/69040/69040_100_10.0.data :
SNIC : k :100 m :10 psnr :19.3731 SizeCompressed : 53428 Recall :0.336772
SLIC : k :100 m :10 psnr :26.0893 SizeCompressed : 81708 Recall :0.843906
```

```

WATERPIXEL : k :100 m :10 psnr :19.6746 SizeCompressed : 50579 Recall :0.293708
./output_stat/69040/69040_100_0.5.data :
SNIC : k :100 m :0.5 psnr :19.8218 SizeCompressed : 54426 Recall :0.327832
SLIC : k :100 m :0.5 psnr :27.7314 SizeCompressed : 102070 Recall :0.942926
WATERPIXEL : k :100 m :0.5 psnr :19.4672 SizeCompressed : 52353 Recall :0.253051
./output_stat/69040/69040_15000_4.0.data :
SNIC : k :15000 m :4 psnr :28.68 SizeCompressed : 225665 Recall :0.999914
SLIC : k :15000 m :4 psnr :33.5639 SizeCompressed : 249156 Recall :1
WATERPIXEL : k :15000 m :4 psnr :23.7068 SizeCompressed : 144148 Recall :0.991233
./output_stat/69040/69040_100_8.0.data :
...

```

A partir de ce fichier, il est facile d'en extraire et trier les données par algorithme et par image. Enfin, il ne reste plus qu'à générer les différentes courbes en fonction du nombre de superpixels et de la métrique à observer (psnr, boundary recall, taille compressée). Pour chaque courbe, une distinction est également faite selon le facteur d'adhésion au contour.

## 1.4 Résultat

Voici un exemple des résultats obtenues par l'algorithme de SLIC qui représente la moyenne sur toutes les images passées à cette algorithme.

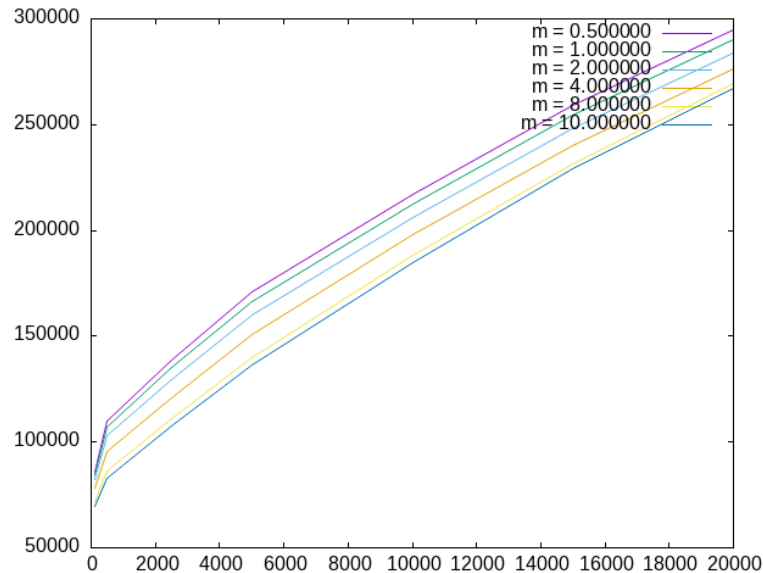


Figure 3: En abscisse :le nombre de super-pixels / En ordonnée: la taille du fichier compressé

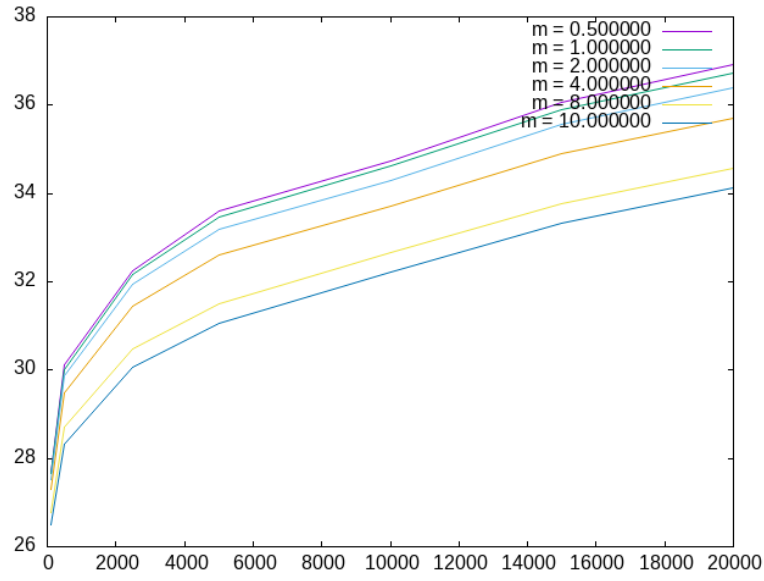


Figure 4: En abscisse : le nombre de super-pixels / En ordonnée: PSNR (en dB)

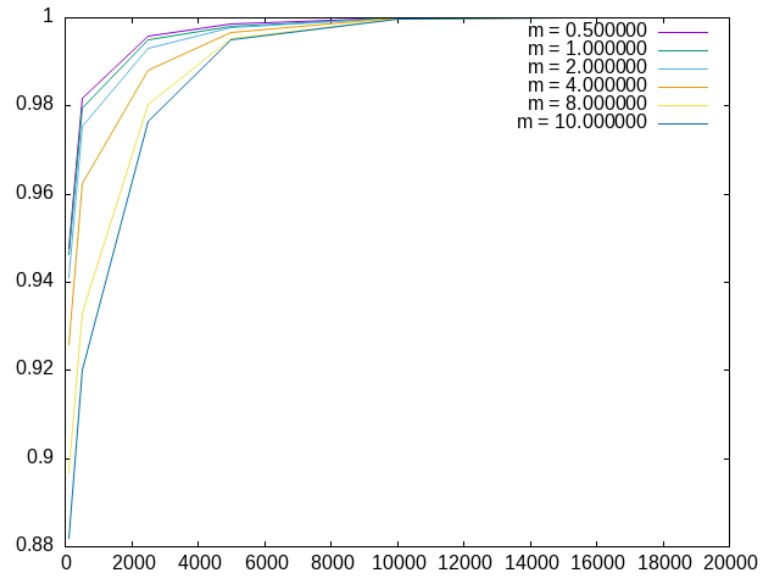


Figure 5: En abscisse : le nombre de super-pixels / En ordonnée: Boundary Recall

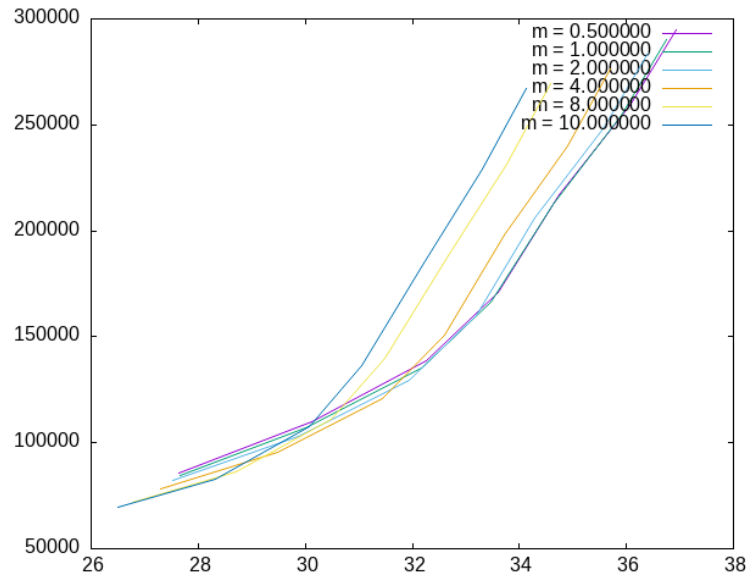
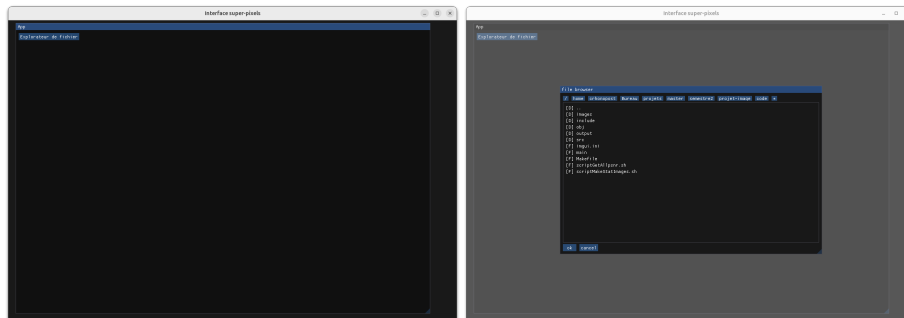


Figure 6: En abscisse : le PSNR (en dB) / En ordonnée: la taille du fichier compressé (en octet)

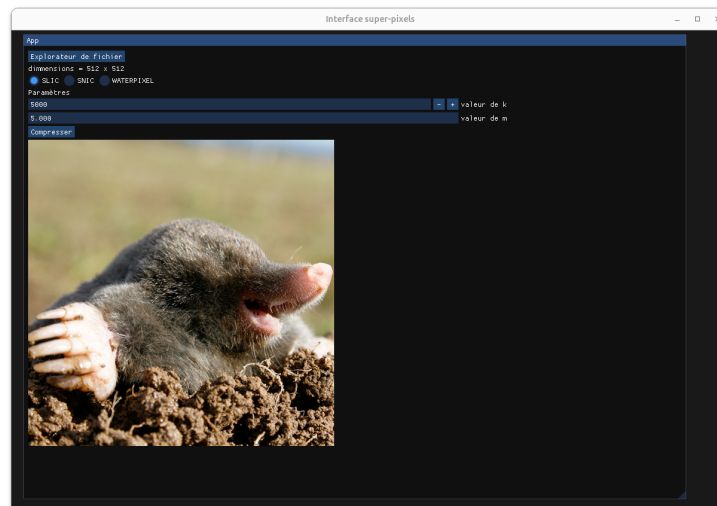
## 2 Interface graphique

Pour l'implémentation de l'interface graphique, nous avons utilisé Dear ImGui, une bibliothèque légère en C++. Son intégration simple et efficace permet de concevoir des interfaces réactives pour modifier des valeurs de variables.

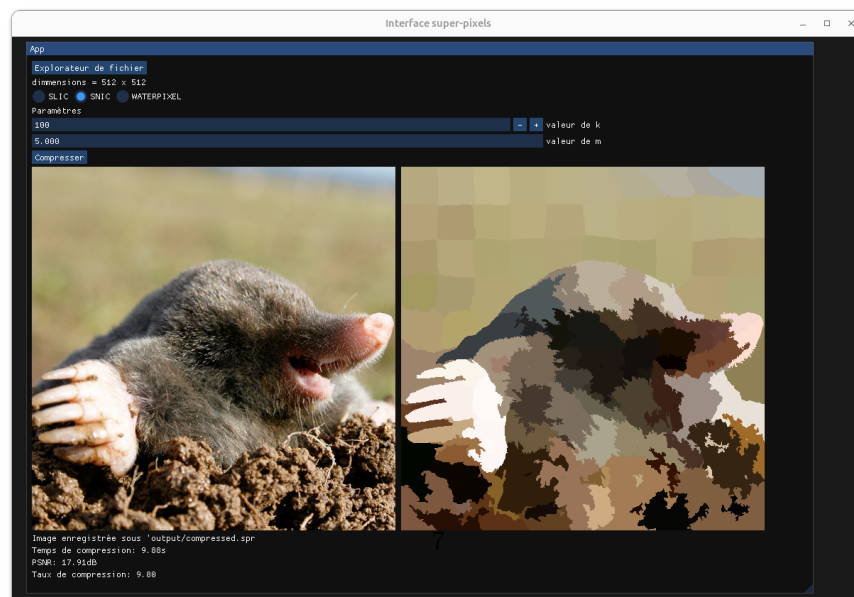


(a) Ecran d'accueil

(b) Explorateur de fichier pour choisir l'image (ne fonctionne qu'avec des ppm)



(c) Paramétrage de la compression pour choisir l'algorithme ainsi que ses paramètres.



(d) Résultat après compression (image compressée à droite) avec des informations sur la compression juste dessous.

Figure 7

## References

- [1] Pablo Arbelaez, Charless Fowlkes et David Martin (2007). *The Berkeley Segmentation Dataset and Benchmark* <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>
- [2] Omar Cornut et contributeurs (2014–présent). *Dear ImGui: Bloat-free Immediate Mode Graphical User interface for C++ with minimal dependencies* <https://github.com/ocornut/imgui>