

Compte Rendu #2 : Implémentation du cœur de l'algorithme SLIC

Mathys CHAMPENOIS, Cesar CHARBEY, Mickael CZOLACZ - Super Pixels

Dépôt GitHub : <https://github.com/Mathys3114/ProjetCCMTI>

1 Bilan de la semaine : Réalisations vs Objectifs

Lors de notre précédent compte rendu, nous avions fixé comme objectif de mettre en place la structure du code C++ et de développer une version stable de l'algorithme SLIC (création de la grille, perturbation du gradient, et boucle K-Means bilatéral) sans l'étape de connectivité.

Cet objectif a été pleinement atteint. Nous disposons aujourd’hui d’une base de code saine, fonctionnelle, et modulaire. L’algorithme parvient à segmenter une image en superpixels et à affecter à chaque région sa couleur moyenne.

2 Architecture du code et Exécutables développés

Pour valider chaque étape de la théorie vue dans l’état de l’art, nous avons divisé notre implémentation en plusieurs programmes de test. Cela nous permet de déboguer visuellement chaque étape mathématique de l’algorithme.

Nous avons créé 4 exécutables distincts basés sur notre bibliothèque partagée (`slic.hpp`) :

1. `genGrid` : Ce programme valide l’étape 1. Il charge une image, calcule le pas $S = \sqrt{N/K}$ et génère la grille régulière de K superpixels. L’image de sortie superpose les centres initiaux sur l’image d’origine.
2. `genGridNudge` : Ce programme ajoute l’étape de perturbation (correction du gradient). Il déplace les centres générés par `genGrid` vers les zones de plus faible gradient dans un voisinage défini par le paramètre g , évitant ainsi de placer les centres sur les contours des objets.
3. `SLIC` : C’est le coeur du projet. Il exécute la boucle complète d’affectation et de mise à jour des centres via le K-Means bilatéral. L’image de sortie est une image où chaque pixel a été remplacé par la couleur moyenne de son superpixel d’appartenance.
4. `SLICPartMap` : Cet utilitaire permet de mieux visualiser la segmentation. Au lieu de remplir les zones, il détecte les frontières entre les superpixels (en comparant les labels des pixels adjacents) et les trace en blanc, produisant une carte de contours.

3 Plan pour la semaine prochaine

Maintenant que le partitionnement K-Means est stable, nous observons le défaut inhérent à SLIC : la présence de pixels isolés (bruit ou superpixels non continus). L'objectif de la Semaine 3 est de régler ce problème :

- **Objectif principal** : Implémenter la phase de Post-Traitement (Connectivité) détaillée dans la section 4 du papier de référence.
- *Résultat attendu* : Un exécutable produisant des cartes de superpixels parfaitement contigus (aucun pixel de bruit).

4 Plan pour les semaines à suivre

Une fois l'algorithme SLIC totalement robuste (connectivité incluse), nous basculerons vers l'objectif final du projet : la Compression.

- **Semaine +2** : Obtenir nos premiers résultats visuels propres et mettre en place une fonction d'évaluation (calcul du PSNR entre l'image reconstruite et l'originale).