

# Segmentation par Superpixels

Tom Sarry

<tom.sarry@umontreal.ca>

Décembre 2022

## 1 Introduction

La segmentation d'une image en plusieurs zones homogènes est un problème intéressant et est utilisée dans de nombreux domaines en informatique comme l'imagerie médicale ou encore la vision par ordinateur. En effet, trouver de nombreuses zones uniformes permet de simplifier l'image, en regroupant les informations redondantes.

La méthode utilisée pour segmenter une image dont nous parlerons sera celle des *Superpixels*: zones homogènes et géographiquement compactes. Plusieurs algorithmes existaient en 2010 afin de segmenter une image de cette manière, mais possédaient soit trop de paramètres, ce qui rendait leur utilisation complexe, ou ne fonctionnaient pas en temps linéaire, propriété cependant désirable.

Nous présentons donc aujourd'hui l'algorithme **SLIC** [1], développé en 2010 par des chercheurs de l'EPFL, fonctionnant en temps linéaire en terme du nombre de pixels de l'image et qui n'a pas besoin de paramètres, à part le nombre de superpixels désirés. Cet algorithme de segmentation nécessite moins de ressources que les autres options à disposition à l'époque, il était donc adapté pour des appareils moins puissants comme des petites caméras ou des téléphones.

## 2 Algorithme

L'algorithme **SLIC**, ou *Simple Linear Iterative Clustering* s'effectue dans l'espace de couleur **LAB** ( $L^*a^*b^*$  CIE 1976<sup>1</sup>), afin de bénéficier de sa propriété d'être perceptuellement uniforme. En effet, plusieurs couleurs paraissant *aussi* différentes par le SVH (Système Visuel Humain), posséderont une distance par norme  $L_2$  similaire.

Cet algorithme utilise donc un espace de dimension 5: chaque pixel est assigné le 5-tuple suivant:  $\langle x, y, l, a, b \rangle$ , qui correspond à sa position  $\langle x, y \rangle$  dans l'image, ainsi que ses paramètres de couleurs dans l'espace **LAB**. Cela nous permet donc de calculer une distance pixel à pixel qui prend en compte l'éloignement physique, mais aussi l'importance de la différence de couleurs entre eux-ci.

La procédure est très similaire à l'algorithme *K-means* et se déroule de la manière suivante, un pseudocode simplifié est également donné dans l'Algorithme 1:

---

<sup>1</sup>[https://fr.wikipedia.org/wiki/L\\*a\\*b\\*\\_CIE\\_1976](https://fr.wikipedia.org/wiki/L*a*b*_CIE_1976)

1. **Initialisation des Centres:** Les  $k$  centres pour les superpixels sont placés à intervalle réguliers pour quadriller l'image initiale.
2. **Ajustement des Centres:** Les centres ont possiblement été positionnés sur des pixels à fort gradient, ie. sur un contour. Il est préférable si c'est le cas de bouger légèrement le centre en choisissant le pixel à plus faible gradient dans son voisinage  $3 \times 3$ .
3. **Répéter jusqu'à convergence ( $E \leq 1$ ):**

- (a) **Assignement des Régions:** Pour chaque pixel, calculer la distance entre celui-ci et tous les centres présents dans son voisinage restreint<sup>2</sup>. Cette distance entre deux pixels  $s$  et  $t$  se fait dans un espace de dimension 5 et se calcule de la manière suivante ( $m$  étant une valeur entre 1 et 20, 10 est recommandé par les chercheurs):

$$D_{s,t} = \sqrt{(x_s - x_t)^2 + (y_s - y_t)^2} + \frac{m}{S} \sqrt{(l_s - l_t)^2 + (a_s - a_t)^2 + (b_s - b_t)^2}$$

Par la suite, on assigne la région de ce pixel comme étant la région du centre avec lequel il avait la plus petite distance, ie. le plus de similarités.

- (b) **Mise à jour des Centres:** Pour chaque centre, mettre à jour ses valeurs  $\langle x, y, l, a, b \rangle$ , en prenant la moyenne des valeurs des pixels contenus dans sa région.
  - (c) **Calcul de la Convergence:** Calculer  $E$ : la norme  $L_1$  entre les précédents centres et ceux venant d'être calculés.
4. **Garantir la Connectivité:** Après avoir atteint convergence, il est possible qu'une région soit disjointe. Nous englobons donc les parties de région qui n'appartiennent pas à la zone *principale* de la région (celle qui inclue le centre de la région), par la région la plus importante qui jouxte cette zone disjointe.

### 3 Discussions

Les auteurs mettent en avant le fait que l'algorithme SLIC n'a besoin que du nombre de superpixels comme paramètre pour fonctionner convenablement. Cependant, il est flexible aux besoins de son utilisateur avec l'ajustement de  $m$ , variable utilisée lors du calcul de la distance dans cet espace 5-dimensions. En effet, en augmentant  $m$ , l'algorithme favorisera des zones plus compactes géographiquement. À l'inverse, réduire  $m$  favorisera des zones de couleurs uniformes.

---

<sup>2</sup>Un carré de  $2S$  pixels de côtés, où si  $N$  est le nombre de pixels de l'image,  $S = \sqrt{\frac{N}{k}}$

---

**Algorithm 1** SLIC

---

```
1: procedure SLIC( $nSuperpixels$ )
2:   Initialize  $nSuperpixels$  centers on a grid at regular interval
3:   Move centers to the lowest gradient in their  $3 \times 3$  neighborhood
4:   repeat
5:     for all cluster centers do
6:       Assign neighboring pixels to this cluster if this center has the lowest
       xylab-distance to them
7:     end for
8:     Compute new cluster centers by taking the average of the  $\langle x, y, l, a, b \rangle$  values
       in each cluster
9:      $E \leftarrow L_1$  distance between old and new cluster centers
10:  until  $E \leq$  threshold  $\triangleright E \leq 1$  in the code
11:  Enforce connectivity
12: end procedure
```

---

## 4 Résultats

Nous présentons dans cette section les résultats obtenus par l'algorithme sur trois images Church, Lady et Fish utilisées dans le papier original, cf Figure 1, 2 et 3.

## 5 Conclusion

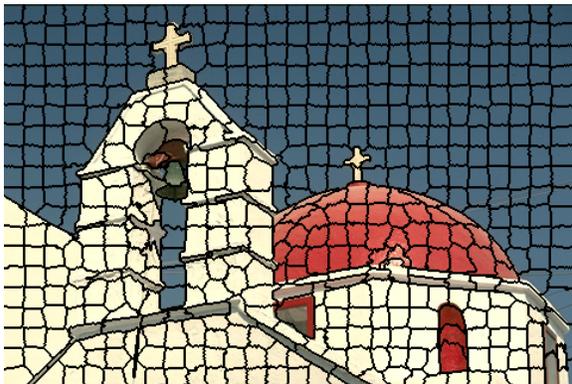
Nous avons présenté dans ce rapport l'algorithme SLIC, permettant de segmenter une image par superpixels, zones géographiquement compactes et de couleur perceptuellement uniforme. Notre implémentation possède un temps de calcul linéaire en termes du nombre de pixels de l'image, et le code pour celle-ci est disponible libre de droits à l'adresse suivante: <https://github.com/tomsarry/slic>.



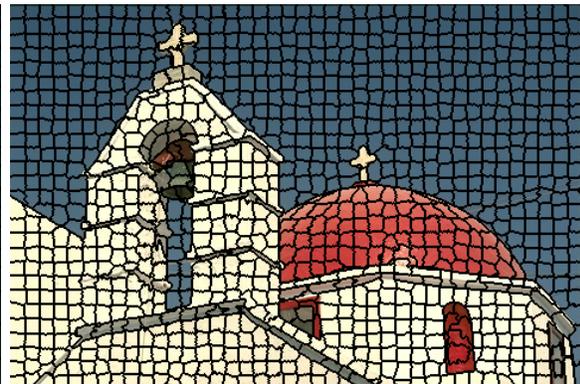
(a) Image originale



(b)  $n = 256$



(c)  $n = 512$

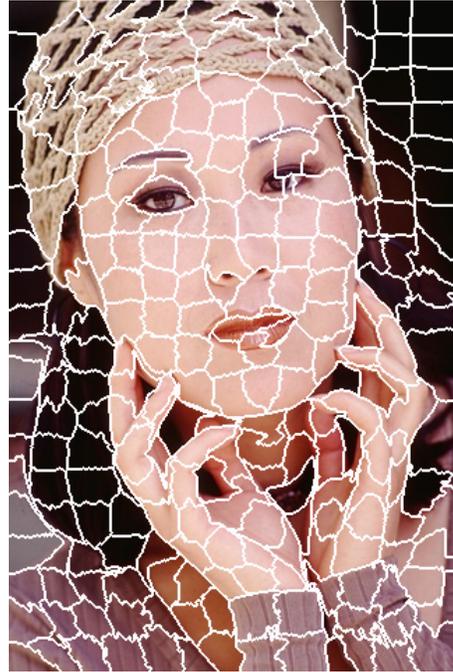


(d)  $n = 1024$

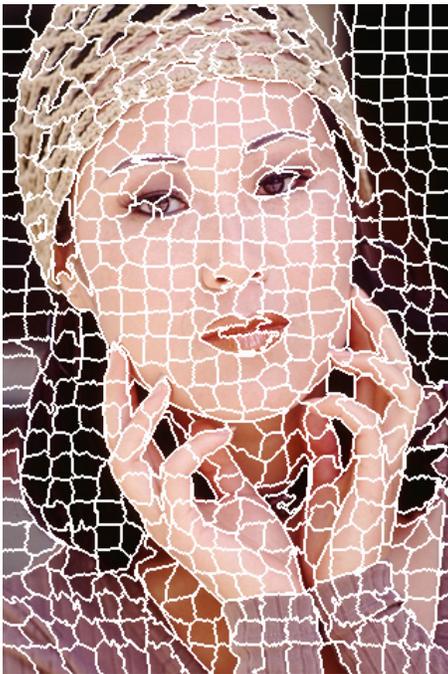
Figure 1: Image originale et segmentée de l'image Church.



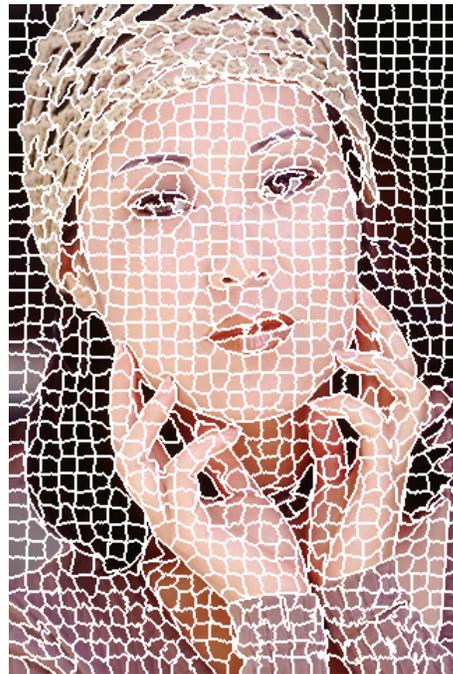
(a) Image originale



(b)  $n = 256$



(c)  $n = 512$



(d)  $n = 1024$

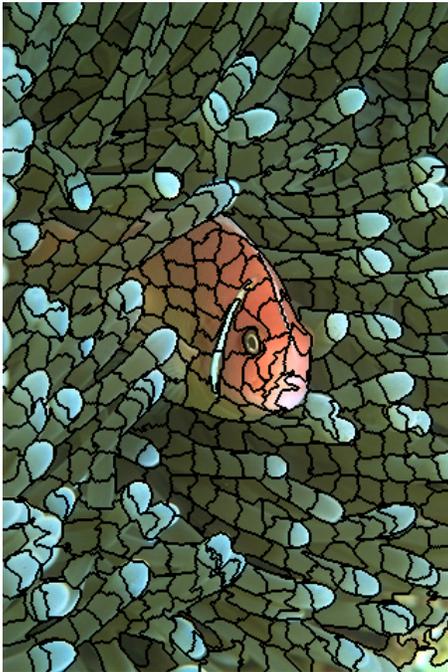
Figure 2: Image originale et segmentée de l'image Lady.



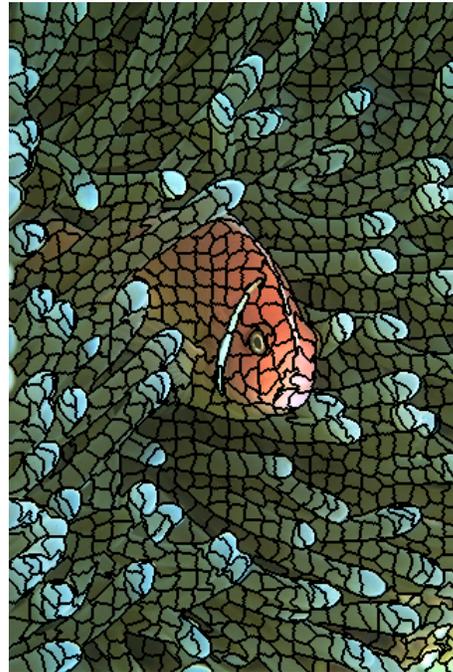
(a) Image originale



(b)  $n = 256$



(c)  $n = 512$



(d)  $n = 1024$

Figure 3: Image originale et segmentée de l'image Fish.

## References

- [1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Ssstrunk, *Slic superpixels*, 2010.