

Rapport de projet IMA

Waterpixels

Ghislain Kengne Gumete - Cédric Vincent

I. Introduction	1
II. Waterpixel	2
A. Calcul du gradient	2
B. Définition des cellules	2
C. Sélection des marqueurs	4
D. Régularisation du gradient	4
E. Watershed	5
III. Évaluation des résultats	5
A. Première analyse des résultats	5
1. Paramètre de régularisation	5
2. Type de maillage (grid)	7
3. Lacunes des résultats	7
B. Trois critères d'évaluation	8
1. Densité de contours (Contour Density)	8
2. Homogénéité (Similarity)	8
3. Adhérence aux contours (Boundary Recall)	8
C. Influence des paramètres	9
1. Filtrage du gradient	9
2. Gradient	9
3. Régularisation k	11
4. Type de maillage	11
D. Comparaison avec SLIC	11
E. D'autres types d'images	13
IV. Conclusion	14
Références	14

I. Introduction

Dans de nombreuses applications en images : segmentation, recalage, etc. il est souvent nécessaire d'avoir une segmentation de plus bas niveau qui sera la base de techniques plus avancées, d'où le besoin d'avoir des superpixels qui segmentent l'image.

Ce projet a pour but la partition d'une image en super-pixels, des zones homogènes respectant les contours de l'image. Cet algorithme en particulier permet de contrôler le nombre de super-pixels et leur régularité, tout en conservant une complexité linéaire et sans post-traitement.

Ce travail est basé sur l'article [1], qui présente les Waterpixels avec leurs différentes variations et évalue leurs performances.

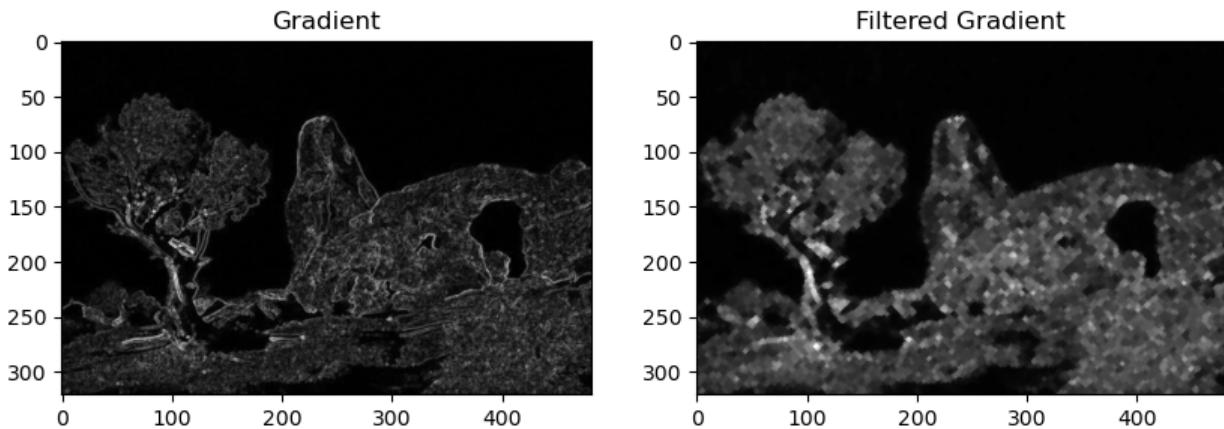
II. Waterpixel

Les principales étapes de l'algorithme :

- Calcul du gradient de l'image
- Définition de cellules régulières dans l'images
- Sélection de marqueurs dans chaque cellule
- Régularisation du gradient avec une carte de distance sur les marqueurs
- Application de l'algorithme du watershed

A. Calcul du gradient

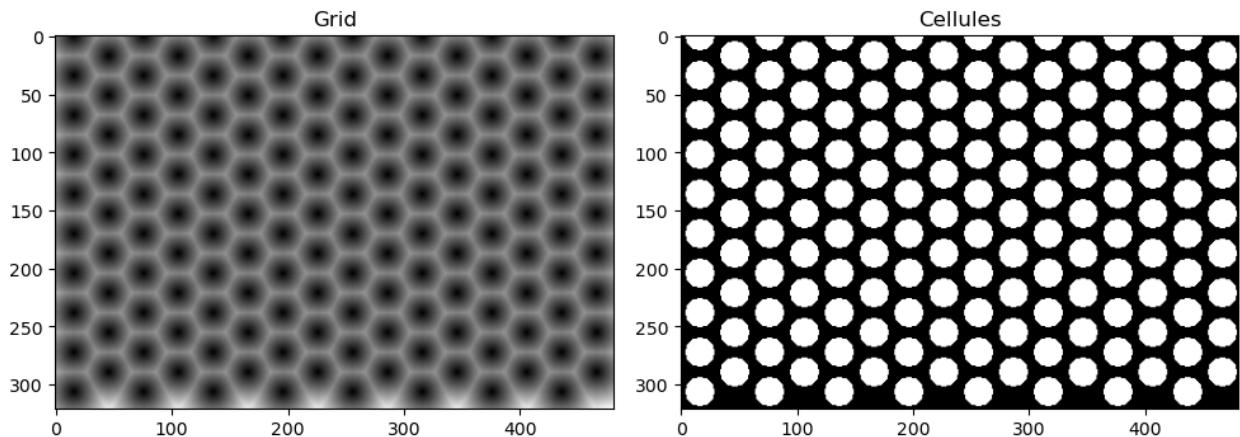
La première étape de l'algorithme consiste à calculer le gradient de l'image. Il existe plusieurs méthodes pour calculer ce gradient : gradient par filtrage comme le filtre de Sobel, gradient morphologique, etc. Initialement, nous utilisons un gradient morphologique avec comme élément structurant un disque de rayon 1. Pour la sélection des marqueurs, nous filtrons le gradient afin de privilégier les zones homogènes pour maximiser les chances d'obtenir des marqueurs adaptés. Le filtre utilisé ici est une ouverture-fermeture, qui permet d'éliminer le bruit et les détails trop petits par rapport aux superpixels.



B. Définition des cellules

Une fois que nous avons le gradient de l'image, nous allons définir les cellules dans l'image. Pour le faire, nous définissons un maillage de points qui seront les centres de chaque

cellules, puis appliquons un algorithme de carte de distance pour obtenir une grille. Deux maillages sont possibles: selon un réseau carré ou hexagonal. Le paramètre contrôlant le nombre de superpixels est *sigma*, le pas du réseau. Enfin, nous appliquons un seuillage pour obtenir des cellules dans une image binaire. La valeur du seuil dépend de la carte de distance obtenue: c'est une fraction de l'intensité maximale de la carte dans un carré central de 2σ de côté. Cela permet de ne pas dépendre de la taille de l'image ou de *sigma*.



L'algorithme de carte de distance que nous avons implémenté est celui de **Chanfrein**. Ce algorithme suit les étapes suivantes :

- Un masque est défini. Nous avons choisi le masque suivant qui approche bien la distance euclidienne.

	11		11	
11	7	5	7	11
	5	0	5	
11	7	5	7	11
	11		11	

- Puis un premier balayage avec le demi masque haut suivant cette formule pour chaque pixel (x, y) et $M(i, j)$ les coefficients du demi masque. Le balayage se fait du haut vers le bas et de gauche à droite.

$$\text{Distance}(x, y) = \min(\text{Distance}(x+i, y+j) + M(i, j))$$

	11		11	
11	7	5	7	11
	5	0		

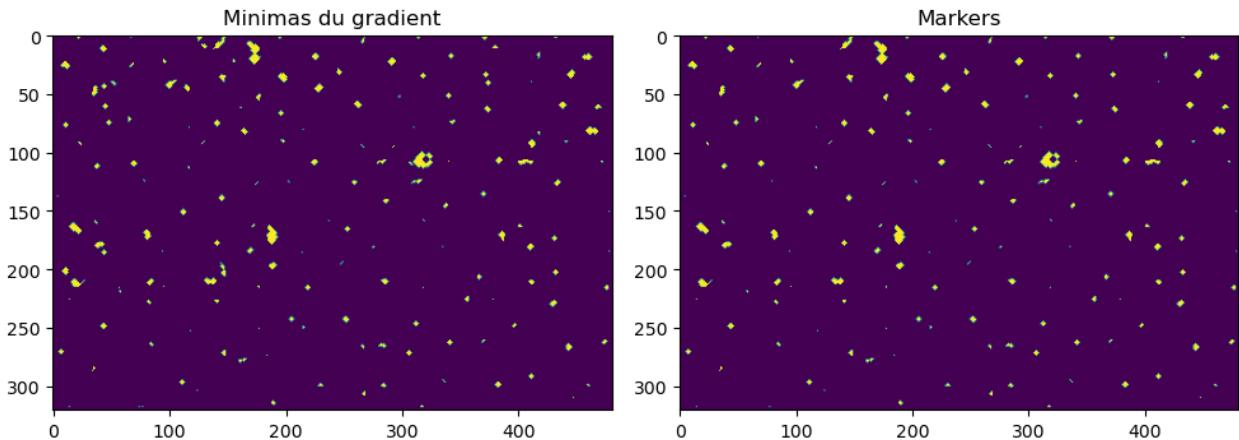
- Enfin un deuxième passage avec le demi masque bas et la même formule. Le balayage se fait du bas vers le haut et de droite à gauche.

		0	5	
11	7	5	7	11
	11		11	

C. Sélection des marqueurs

La partition de l'image en cellules permet de définir les marqueurs qui seront les points de départ du Watershed. Le but est d'obtenir un marqueur par cellule, qui sera l'origine du superpixel, et qui correspond au point le plus bas pour débuter le "flooding" de l'algorithme du Watershed. Ces marqueurs sont donc des minima du gradient de l'image. On cherche ainsi, dans chaque cellule, le plus grand groupe de points connexes avec comme valeur le minimum du gradient dans la cellule.

Pour cela, on commence par calculer, dans chaque cellule, tous les points au minimum du gradient. En utilisant le même algorithme d'étiquetage en composantes connexes que précédemment, toujours dans chaque cellule, on en déduit les marqueurs.

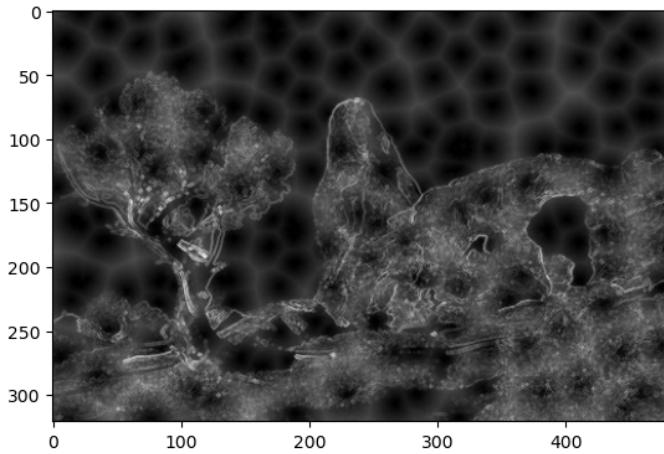


D. Régularisation du gradient

Avant d'appliquer la transformation du watershed à partir des marqueurs, nous devons nous assurer de la régularité des superpixels. Pour cela, nous régularisons spatialement le gradient de l'image à partir des marqueurs. En calculant une carte de distance à partir de ces marqueurs, nous obtenons des frontières possibles entre les superpixels en l'absence de contours dans l'image. Cette carte de distance est normalisée par une multiplication avec $\frac{2}{\sigma}$, ce qui rend la régularisation indépendante de σ , c'est-à-dire du nombre de superpixels. Il s'agit enfin d'ajouter cette carte au gradient de l'image, avec un facteur de régularisation k .

Avec g le gradient de l'image et $d_{marqueurs}$ la carte de distance obtenue:

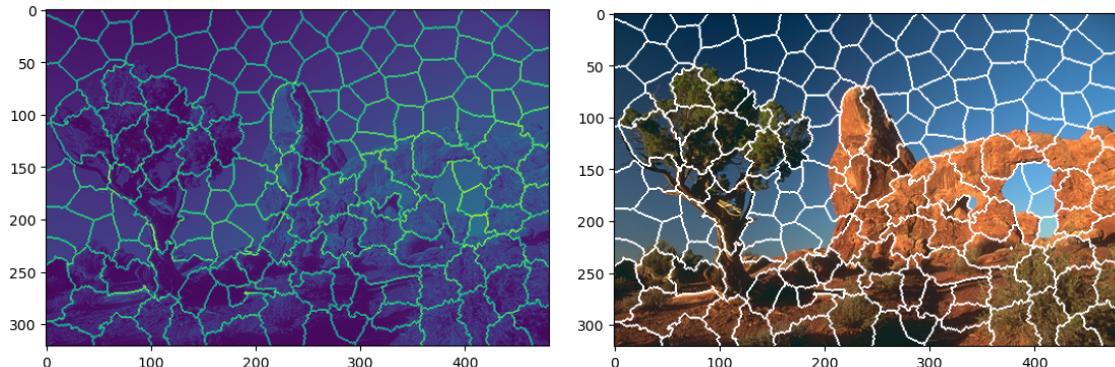
$$g_{reg} = g + k \times d_{marqueurs}$$



Régularisation du gradient avec $k=8$

E. Watershed

La transformation du watershed est finalement appliquée au gradient régularisé à partir des marqueurs. L'implémentation utilisée est celle de scikit-image. On peut ensuite visualiser le résultat en ajoutant les bordures sur l'image en noir et blanc. Les bordures sont les pixels frontaliers entre deux régions (les pixels connexes à un pixel de label différent, en 4-connexité), elles font donc 2 pixels de large. Nous les utilisons au lieu de la ligne de partage des eaux du watershed à fin de faciliter la comparaison avec les SLIC: les deux algorithmes renvoient ainsi une partition de l'image en régions sans pixels de frontières.

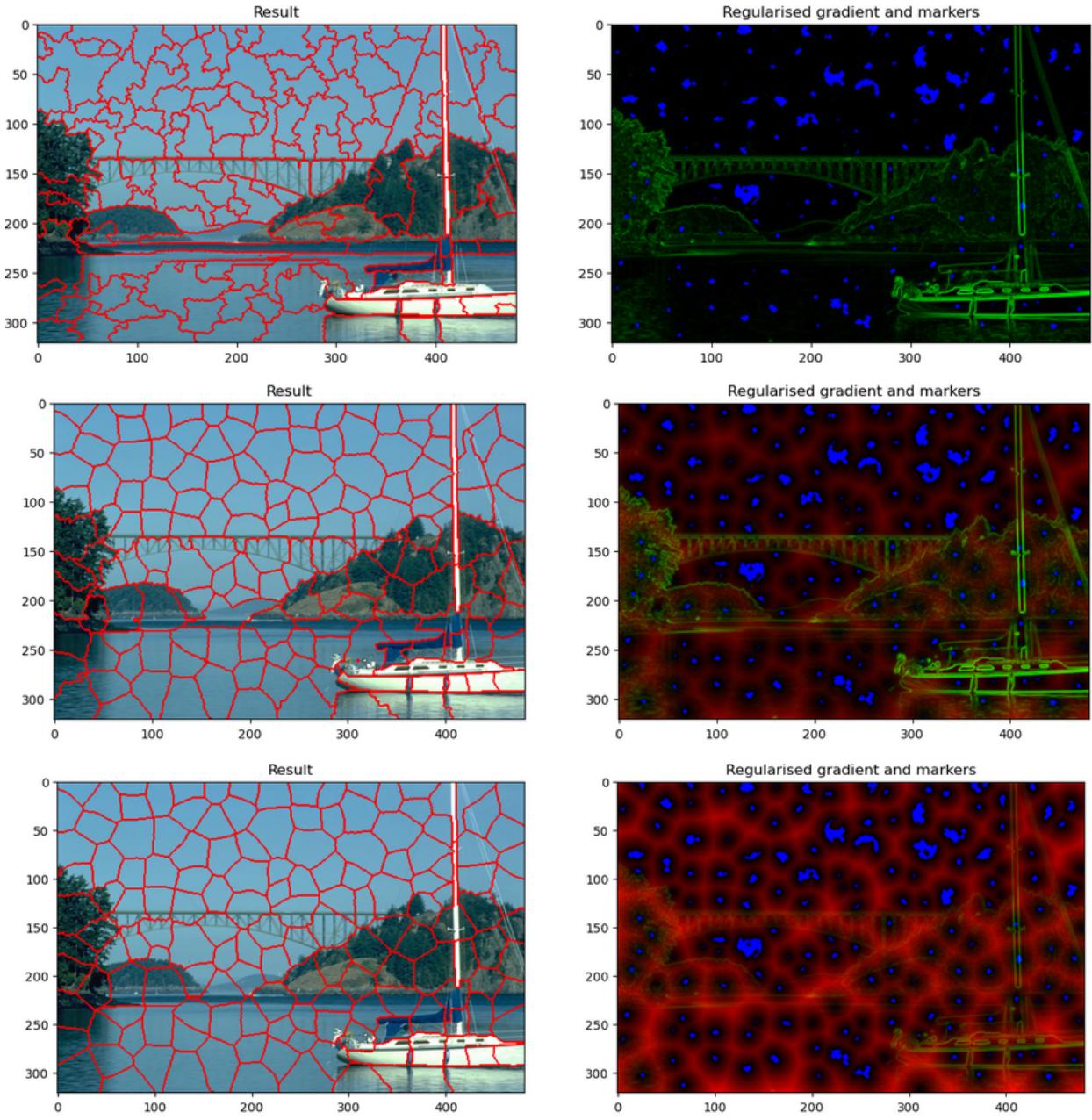


Visualisation des bordures obtenues

III. Évaluation des résultats

A. Première analyse des résultats

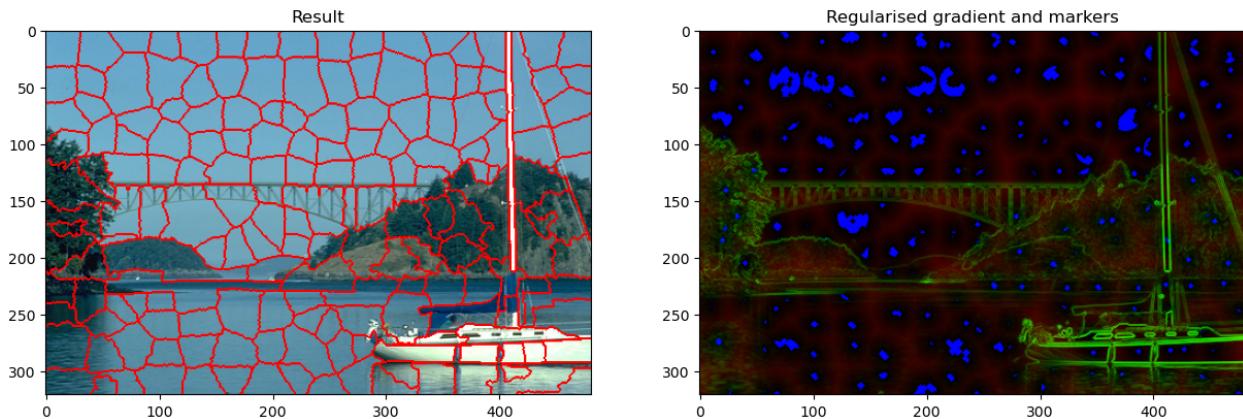
1. Paramètre de régularisation



Résultats pour $k=0, 8$ et 32 , avec gradient régularisé

Avec $k=0$, Il n'y a aucune régularisation dans la taille et la forme des superpixels, qui suivent simplement les bordures trouvées par le gradient. Augmenter k permet d'avoir des formes plus régulières, de tailles relativement homogènes. Lorsque k est suffisamment grand, le gradient devient négligeable face à la carte de distance des marqueurs, et on obtient bien la partition de Voronoï des marqueurs.

2. Type de maillage (grid)

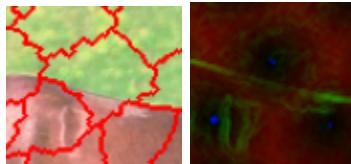


Résultats pour k=8 et maillage carré, avec gradient régularisé

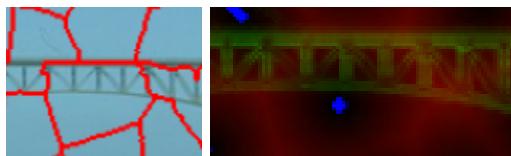
Les marqueurs obtenus sont différents, et la régularité des superpixels semble moins bonne qu'avec un maillage hexagonal.

3. Lacunes des résultats

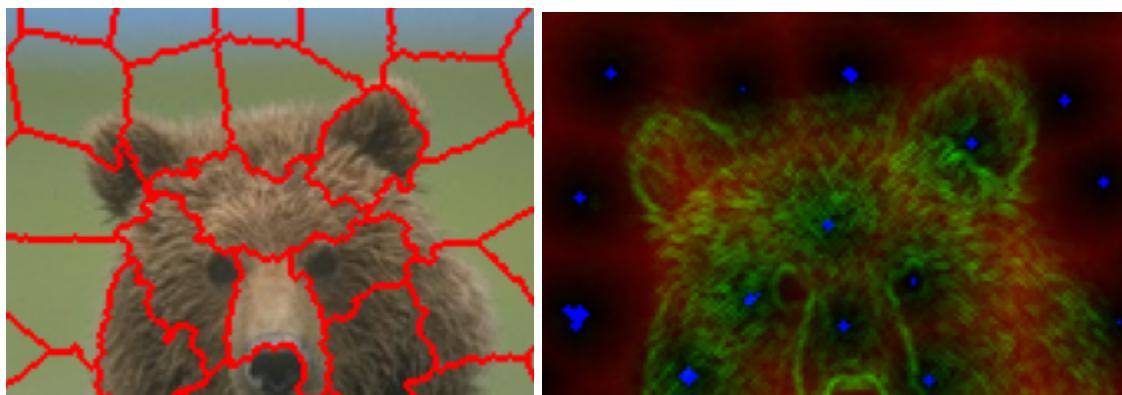
On cherche ici à comprendre pourquoi des contours sont ignorés par notre algorithme. Ces erreurs peuvent être dues à des lacunes dans le gradient:



Lorsque des marqueurs sont trop proches des contours, ils peuvent les ignorer:



Un gradient peu prononcé à l'endroit où on voudrait la segmentation pose problème:



Nous allons désormais chercher à analyser de manière quantitative les résultats.

B. Trois critères d'évaluation

L'article nous donne trois critères d'évaluation des résultats:

1. Densité de contours (Contour Density)

Avec n et m les dimensions de l'image, S_c les pixels dans les bordures (divisées par 2 car elles sont 2 pixels de large) et S_b les pixels dans les bords de l'image:

$$CD = \frac{\frac{1}{2}|S_c| + |S_b|}{n*m}$$

CD représente le nombre de pixels nécessaires pour représenter les bordures des superpixels. C'est donc une mesure de la régularité de leurs contours, sous la forme de la complexité de chaque superpixel. On cherche donc à minimiser cette quantité, afin d'obtenir des superpixels plus simples et plus réguliers.

2. Homogénéité (Similarity)

Avec N le nombre de superpixels:

$$MF = \frac{1}{N} \sum_{i=1}^N mf(s_i^*, s_{avg}^*)$$

On définit mf comme le *mismatch factor*, c'est-à-dire la proportion de pixels qui diffèrent entre les deux formes. s_i^* est le superpixel i centré dans l'image, et s_{avg}^* est la forme moyenne des superpixels centrés.

MF est donc une mesure de la régularité de la forme et de la taille des superpixels, indépendante de leur forme générale (rectangulaire ou hexagonale par exemple). On cherche ainsi à le minimiser.

3. Adhérence aux contours (Boundary Recall)

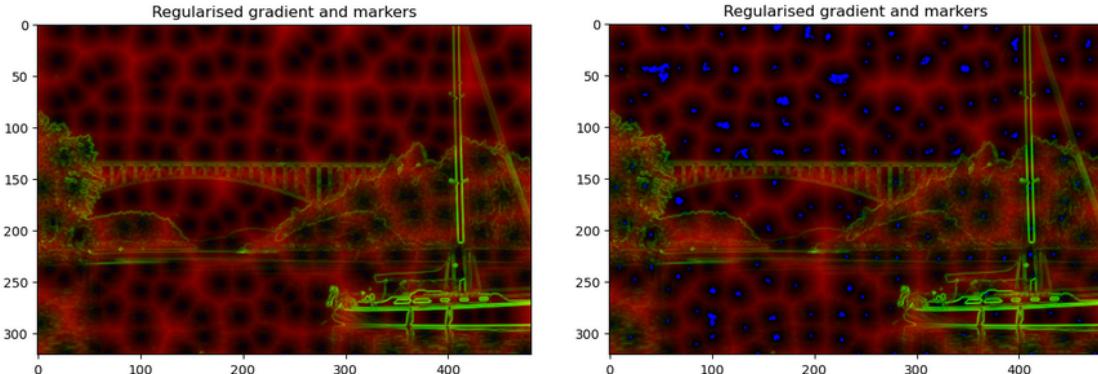
Finalement, il reste à mesurer la précision du respect des contours de l'image. Pour cela, il est nécessaire d'avoir une segmentation de référence. On évalue ainsi la proportion des pixels de cette segmentation étant suffisamment proches de contours de superpixels:

$$BR = \frac{|\{p \in Segmentation, d(p, Contours) < 3\}|}{|Segmentation|}$$

L'objectif est donc de maximiser BR. Or avoir des superpixels avec des contours plus sinués permet d'avoir plus de pixels dans les contours, et donc un BR plus élevé, au détriment de la régularité des superpixels. Un compromis apparaît donc entre les différents critères.

C. Influence des paramètres

1. Filtrage du gradient



Marqueurs et régularisation obtenues avec filtre gaussien ($\sigma=2$, à gauche) et ouverture/fermeture (avec un disque de rayon 1)

Les opérations morphologiques d'ouverture et fermeture, qui effacent les détails, font apparaître de plus gros marqueurs (en bleu) qu'un filtre gaussien, mais la régularisation du gradient reste très similaire. On en déduit que l'emplacement des marqueurs reste le même dans les deux cas. Ce filtrage est seulement utilisé pour le calcul des marqueurs et non pas pour le Watershed.

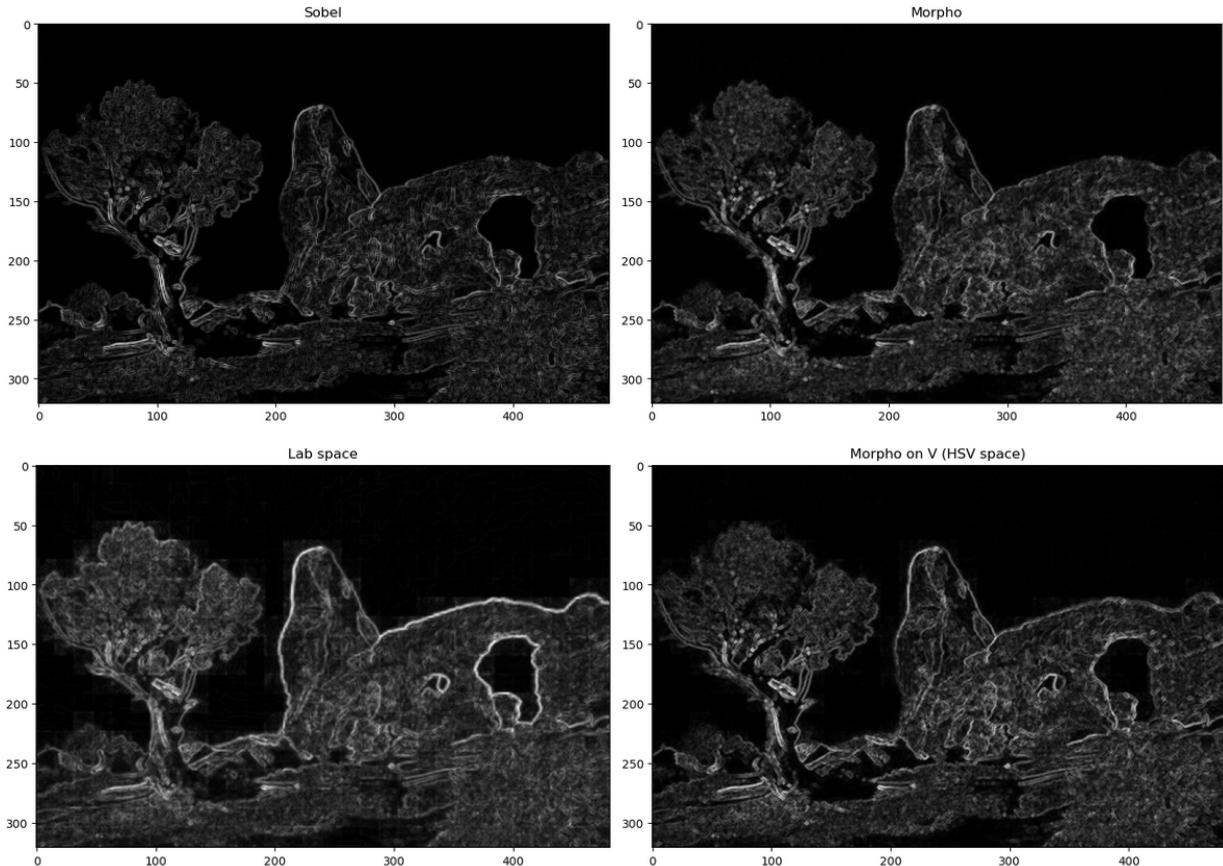
Les tests sont effectués sur un échantillon de [2] (10 images) et moyennés.

$\sigma=20, k=8$	Contour Density	Mismatch Factor	Boundary Recall
Gaussian ($\sigma=2$)	0.069	0.664	0.646
Opening-closing (disque de rayon 1)	0.069	0.673	0.632

Avec un MF supérieur et un BR inférieur, le second filtre donne de moins bonnes performances que le premier. On conclut qu'avoir des marqueurs trop imposants comme points de départ du watershed déséquilibre la régularité des superpixels sans apporter de bénéfices.

2. Gradient

Nous avons pu constater que beaucoup d'erreurs d'adhérences aux contours étaient dues au gradient ne mettant pas ces contours en valeur. Pour améliorer cet aspect, qui revient à augmenter BR, il faudrait améliorer le gradient. Nous avons donc tester plusieurs gradient, en cherchant notamment à lui donner une sensibilité aux changement de couleur. En utilisant la Value de l'espace de couleur HSV, on obtient de meilleur résultats qu'avec l'image convertie en gris. La meilleure option semble être de convertir l'image dans l'espace de couleur Lab, d'effectuer un gradient (morphologique ici) sur chacune des composantes et de les sommer.



$\sigma=20, k=8$	Contour Density	Mismatch Factor	Boundary Recall
Sobel	0.070	0.676	0.645
Morphologique	0.069	0.664	0.646
Morph. Sur V (HSV)	0.070	0.665	0.679
Somme des Morph. sur Lab	0.070	0.658	0.681

La densité de contours n'est pas impactée par ce paramètre, on s'intéresse donc au compromis entre MF et BR. La somme des gradients dans l'espace Lab donne les meilleurs résultats, avec des superpixels plus réguliers adhérents mieux aux contours.

Toutefois, ces résultats dépendent sûrement des images traitées et donc de la base de données considérée. Ces paramètres seront ceux utilisés pour la suite.

3. Régularisation k

Le paramètre de régularisation k a évidemment une influence sur la régularité des superpixels, il s'agit donc de savoir quelle valeur de k donne le meilleur compromis entre la régularité et l'adhérence aux contours:

	Contour Density	Mismatch Factor	Boundary Recall
Waterpixels, k=0	0.079	0.836	0.766
Waterpixels, k=8	0.070	0.658	0.681
Waterpixels, k=16	0.067	0.612	0.591

Si on considère le rapport entre MF et BR comme une mesure de ce compromis, alors k=8 donne les meilleurs résultats.

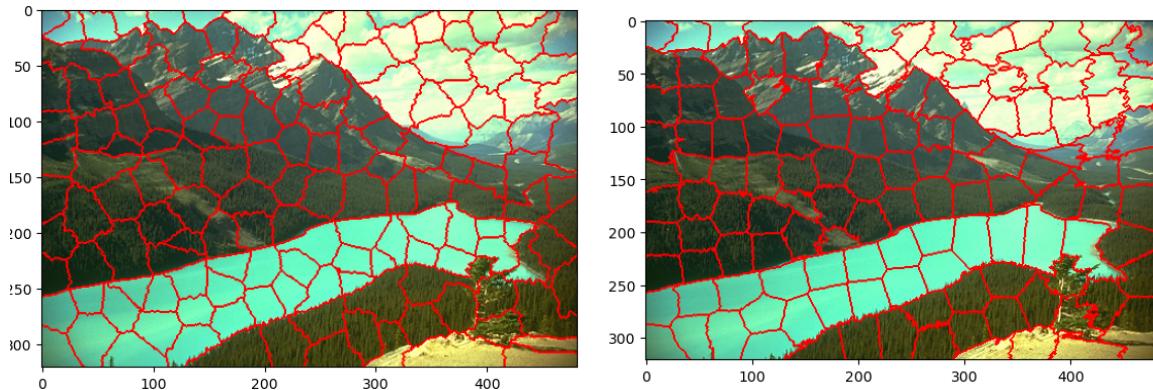
4. Type de maillage

k=8	Contour Density	Mismatch Factor	Boundary Recall
Hexagonal	0.070	0.658	0.681
Square	0.077	0.731	0.733

Comme intuité par nos observations, un maillage carré résulte en des superpixels moins réguliers qu'un maillage hexagonal, comme le montre les valeurs de CD et MF. L'adhérence aux contours est supérieure, toutefois cela peut être simplement dû aux bordures plus sinuueuses des superpixels carrés: le maillage hexagonal offre un meilleur compromis entre BR et CD que le maillage carré.

D. Comparaison avec SLIC

Pour la comparaison avec cet autre algorithme de partition d'images en superpixels (Simple Linear Iterative Clustering), nous utilisons l'implémentation de scikit-image (`skimage.segmentation.slic`), avec le même nombre de superpixels que notre algorithme, et *compactness* comme paramètre de régularisation.

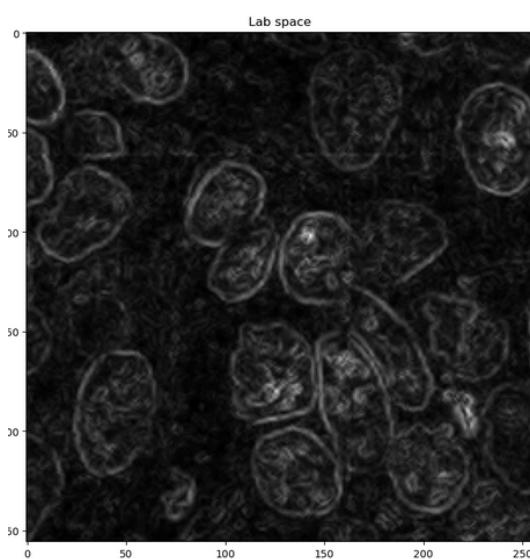
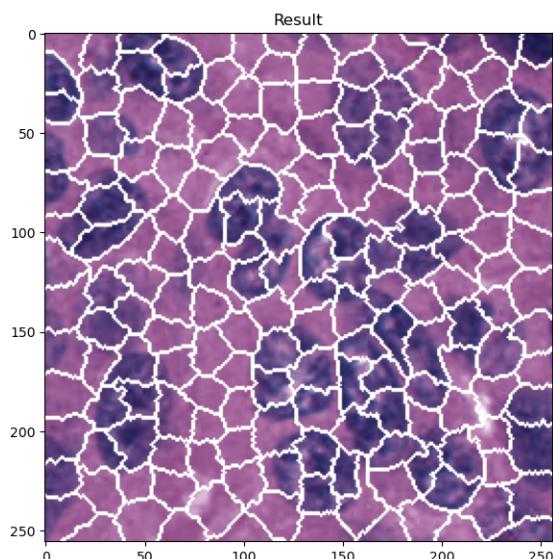
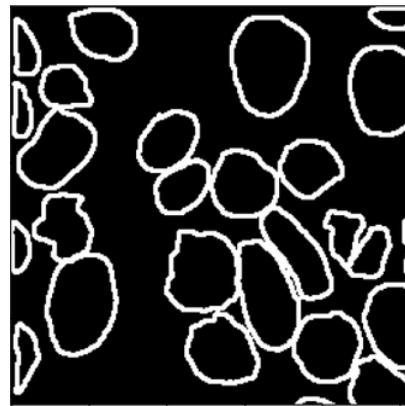
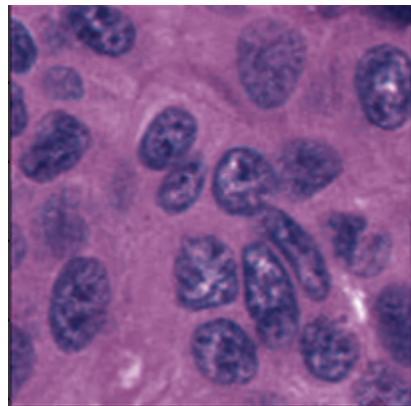


A gauche, les Waterpixels avec $k=8$, et à droite SLIC avec compactness=20

	Contour Density	Mismatch Factor	Boundary Recall
Waterpixels, $k=0$	0.079	0.836	0.766
Waterpixels, $k=8$	0.070	0.658	0.681
Waterpixels, $k=16$	0.067	0.612	0.591
SLIC, no compactness	0.080	0.696	0.681
SLIC, $c=20$	0.076	0.625	0.723
SLIC, $c=100$	0.065	0.326	0.488

SLIC est aussi capable de varier le compromis entre régularité et adhérence aux contours, en ayant une régularité très variable selon les régions de l'image. Les performances entre les deux algorithmes sont comparables, même si l'implémentation de SLIC semble donner de meilleurs résultats que la nôtre (avec $k=8$ et compactness=20, en regardant les rapports entre BR et MF, par exemple).

E. D'autres types d'images



	Contour Density	Mismatch Factor	Boundary Recall
Waterpixels, k=8	0.107	0.611	0.603
SLIC, c=10	0.120	0.728	0.772

Les paramètres utilisés précédemment sont adaptés à la base de données utilisée, et ne sont sûrement pas les plus adaptés ici. Plus précisément, le gradient utilisé rend difficile l'adhérence aux contours.

IV. Conclusion

Nous avons implémenté un algorithme de Waterpixels avec des performances comparables à celle de SLIC, malgré une implémentation plus lente. Les résultats sont dépendants des différents paramètres: le choix du gradient, son filtrage pour les marqueurs et le paramètre de régularisation k . Les deux premiers sont dépendants des images traitées tandis que k peut être optimisé pour donner le meilleur compromis entre régularité des superpixels et adhérence aux contours. Ainsi, une piste pour l'automatisation du choix des paramètres serait de calculer les trois critères d'évaluations pour optimiser ce compromis, c'est-à-dire le compromis entre CD et BR et MF et BR.

Références

[1]: Waterpixels, Vaïa Machairas

<https://hal.archives-ouvertes.fr/hal-01212760/document>

[2]: Berkeley Segmentation Data Set and Benchmarks 500, Contour Detection and Hierarchical Image Segmentation, P. Arbelaez, M. Maire, C. Fowlkes and J. Malik., IEEE TPAMI, Vol. 33, No. 5, pp. 898-916, May 2011

[3]: Cours de Anne Vialard, université Bordeaux 1, Segmentation et Analyse d'images

<https://dept-info.labri.fr/~vialard/Image3D/enonces/tp-watershed.pdf>

[4]: Etiquetage/Analyse en composantes connexes, Laurent Cabaret, Coté recherche