



«Waterpixels» : des superpixels respectant les contours des objets dans les images

Encadré par :

Pr. BLOCH Isabelle

Réalisé par :

BENAISSE Rania

22/05/2022

TABLE DES MATIÈRES

Introduction	1
1. État de l'art	2
2. Méthode des Waterpixels	3
2.1 Calcul du gradient de l'image	4
2.2 Définition de cellules régulières	5
2.3 Sélection des marqueurs	6
2.4 Régularisation spatiale du gradient	8
2.5 Algorithme de la ligne de partage des eaux	9
3. Implémentation et choix des paramètres	10
3.1 Calcul du gradient	11
3.2 Définition de la carte distance	13
3.3 Taille de la grille hexagonale	14
3.4 Facteur de régularisation k	15
4. Évaluation de la méthode	16
4.1 Adhérence aux contours des objets	17
4.2 Régularité des superpixels	20
4.3 Efficacité de l'implémentation	22
5. Application de bureau	23
Conclusion	25

LISTE DES FIGURES

1	Exemple de calcul du gradient sur une image de la base BSDS500	5
2	Exemple de grilles hexagonales selon différentes valeurs du pas de la grille p et du facteur d'homothétie ρ	6
3	Quelques résultats de la sélection de marqueurs avec le pas de la grille $p = 40$ et un lissage gaussien d'écart-type $\sigma = 1$	7
4	Résultat de la régularisation spatiale du gradient en utilisant l'opérateur de Sobel avec un lissage gaussien d'écart-type $\sigma = 1$, le pas de la grille $p = 40$ et un facteur de régularisation $k = 4$	9
5	Waterpixels obtenus en utilisant l'opérateur de Sobel avec un lissage gaussien d'écart-type $\sigma = 1$, un pas de grille $p = 40$ et un facteur de régularisation $k = 4$.	10
6	Formulation de la norme du gradient selon l'opérateur de Sobel	11
7	Waterpixels obtenus en utilisant l'opérateur de Sobel avec un lissage gaussien d'écart-type σ et le gradient morphologique en fixant le pas de grille p à 30 et le facteur de régularisation k à 4	12
8	Zoom sur les Waterpixels obtenus en utilisant l'opérateur de Sobel avec le pas de grille $p = 30$ et le facteur de régularisation $k = 4$	13
9	Illustration du masque des distances locales	14
10	Waterpixels obtenus en variant le pas de grille p avec le facteur de régularisation $k = 10$ et un écart-type $\sigma = 0.3$ pour le lissage gaussien	15
11	Waterpixels obtenus avec $p = 40$ et faisant varier le facteur de régularisation k et un écart-type $\sigma = 0.3$ pour le lissage gaussien	16
12	Exemple d'images de la base BSDS500 avec leurs vérité terrain	17
13	Exemple d'images de la base SBD avec leur vérité terrain	17
14	Résultats de l'évaluation moyenne de l'adhérence des superpixels aux contours des objets avec l'écart type $\sigma = 1$	19

15	Résultats de l'évaluation moyenne de la régularité globale en fonction du nombre de superpixels avec l'écart type $\sigma = 1$ et différents pas de grille p	21
16	Résultats de l'évaluation moyenne du temps d'exécution de notre implementation	23
17	Interface de chargement de l'image et de sélection des paramètres	24
18	Interface de visualisation des Waterpixels	24

Introduction

Les superpixels résultent du partitionnement d'une image en de petites régions homogènes dont l'enjeu est de maintenir une faible complexité calculatoire tout en préservant un ensemble de propriétés telles que:

- La régularité des régions et la connectivité entre elles.
- L'adhérence aux contours des objets de l'image.
- Le contrôle sur le nombre de superpixels générés.

Cette opération dite de «*bas niveau*» connaît un succès grandissant de par sa nature qui permet de réduire efficacement le nombre de pixels à traiter à des primitives que l'on peut directement employer dans des traitements plus poussés en analyse d'images notamment lors de la segmentation, de la détection d'objets et de la classification d'images.

Dans le cadre de mon projet de recherche, je m'intéresserai particulièrement à la génération de superpixels par la méthode des «*Waterpixels*» proposée par *Machairas et al.*[1] qui repose sur la segmentation par ligne de partage des eaux. L'objectif sera de comprendre, d'implémenter et d'améliorer cette méthode puis d'évaluer ses performances en fonction de différentes métriques, selon différents jeux de données et en faisant varier ses hyperparamètres. Je conclurai ma recherche avec une étude comparative entre la méthode des «*Waterpixels*» et d'autres méthodes de complexité linéaire les plus connues dans la littérature.

1 État de l'art

Les superpixels ont été introduit en 2003 par *Ren et Malik* [2] dans le cadre de la segmentation d'images. Depuis, leur adoption s'est élargit à divers domaines d'application entre autres: le tracking, la reconstruction 3D, la détection d'objets... et beaucoup de travaux de recherche se sont axés sur l'implementation de nouveaux algorithmes de génération de superpixels.

Dans leur article, Stutz et al [3] proposent une classification des récents algorithmes de superpixels en fonction de leurs approches :

- **Les algorithmes basés sur la ligne de partage des eaux :** ils se caractérisent par une phase de sélection de marqueurs que l'on donne en entrée à un algorithme de la ligne de partage des eaux pour produire autant de superpixels qu'il y a de marqueurs. De plus, certains algorithmes tels que les Waterpixels ou les Compact Watershed [4] garantissent leur régularité.
- **Les algorithmes basés sur la densité:** le plus connu de cette catégorie est «*quick shift*» [5] dont le principe est d'assigner à chaque pixel le mode correspondant trouvé lors du calcul de la densité d'une image. Néanmoins, ces algorithmes ne permettent pas de préalablement fixer le nombre de superpixels et ne garantissent pas leur régularité.
- **Les algorithmes basés sur les graphes:** considèrent l'image comme un graphe non-orienté où chaque pixel est un noeud du graphe que l'on partitionne sur la base de poids donnés aux contours présents dans l'image. Ces algorithmes se distinguent les uns des autres par l'algorithme de partitionnement utilisé.
- **Les algorithmes basés sur l'évolution des contours:** reposent sur des algorithmes de croissance de régions où les superpixels ne sont autres que des contours qui évoluent à partir d'un ensemble de pixels appelés «*graines*».
- **Les algorithmes basés sur les chemins:** ces algorithmes connectent des graines en construisant un chemin de pixels satisfaisant des critères spécifiques (connectivité, taille des contours, homogénéité des couleurs...). Ils n'assurent pas la régularité des superpixels mais permettent d'en fixer le nombre.

- **Les algorithmes basés sur le clustering:** s'appuient sur des algorithmes de clustering où une sélection de graines de départ est faite et à partir desquelles des clusters sont formés selon certaines caractéristiques telles que la couleur ou les informations spatiales sur les pixels.
- **Les algorithmes basés sur l'optimisation d'énergie:** les pixels sont initialement répartis en superpixels sur la base d'une grille régulière. Puis à chaque itération, l'affection des pixels aux superpixels évolue par rapport à une fonction d'énergie que l'on cherche à optimiser. L'algorithme SEEDS [6] (Superpixels Extracted via Energy- Driven Sampling), par exemple, cherche à optimiser une fonction d'énergie définie par deux termes: la vraisemblance des couleurs des superpixels et un terme de contours qui pénalise leurs irrégularité. L'avantage de ces algorithmes est que l'on peut contrôler la régularité et le nombre de superpixels générés.

2 Méthode des Waterpixels

Pour une image donnée, produire des superpixels par la méthode des waterpixels revient à considérer l'algorithme décrit par les étapes suivantes:

1. Calcul du gradient de l'image
2. Définition de cellules régulières à partir d'une grille régulière
3. Sélection des marqueurs dans chaque cellule de la grille
4. Régularisation spatiale du gradient de l'image en définissant une carte distance à partir des marqueurs précédemment sélectionnés
5. Application de l'algorithme de la ligne de partage des eaux sur le gradient régularisé

Ainsi, on détaillera chaque phase dans les sous-sections ci-dessous.

2.1 Calcul du gradient de l'image

La première étape de l'algorithme revient à calculer le gradient de l'image. Comme ce calcul dépend fortement du type d'images utilisées, on considérera deux opérateurs fréquemment appliqués sur des images en niveau de gris : l'opérateur de Sobel et le gradient morphologique.

- **L'opérateur de Sobel:** le gradient résultant de l'opérateur de Sobel n'est autre que la norme des approximations des dérivées horizontales et verticales de l'image obtenues par convolution avec les filtres de Sobel. Cette opération est souvent précédée par un lissage gaussien d'écart-type σ afin de réduire le bruit qui en résulte.
- **Le gradient morphologique:** est défini comme étant la différence entre la dilatation et l'érosion morphologique appliqués à l'image avec un élément structurant fixe.

L'exemple de la figure 1 illustre les résultats du calcul du gradient en fonction des opérateurs pris en compte que l'on définira plus en détails dans la section 3.

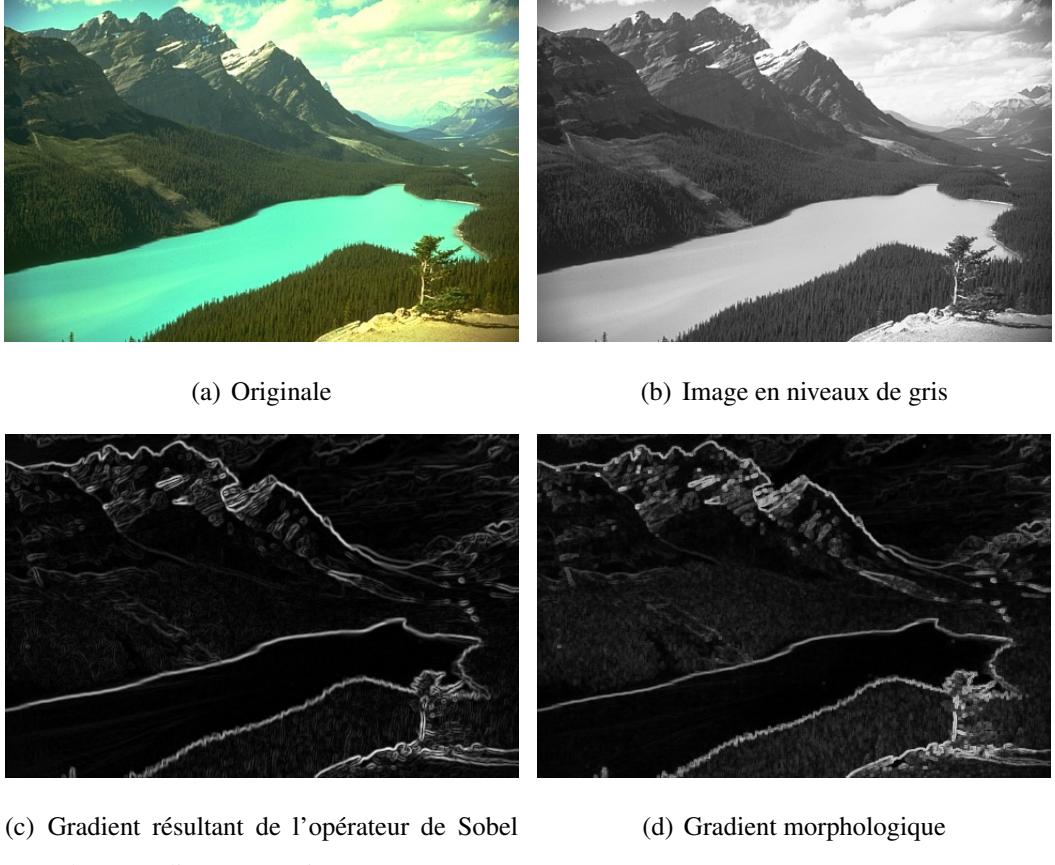


Figure 1: Exemple de calcul du gradient sur une image de la base BSDS500

2.2 Définition de cellules régulières

Comme suggéré dans l'article [1], on choisit de définir les cellules régulières de formes hexagonales. Ces cellules serviront plus tard à déterminer les marqueurs (ou centres de cellules) donnés en entrée à l'algorithme de la ligne de partage des eaux.

La taille ainsi que le nombre de cellules peuvent être réglées à l'aide du pas de la grille qui n'est autre que le rayon d'une cellule que l'on notera p .

Pour chaque cellule C , on définit une marge à partir de la distance entre les sommets de C et les sommets de la cellule obtenue en appliquant une homothétie centrée sur C de facteur ρ (avec $\rho \in]0, 1]$). Comme le facteur d'homothétie ρ contrôle la taille de la cellule obtenue par homothétie centrée sur C , si ρ s'éloigne de 1, la cellule obtenue rétrécit et la marge devient de plus en plus grande. Dans le cas où ρ est égal à 1 aucune marge n'est ajoutée, comme le montre

la figure 2.

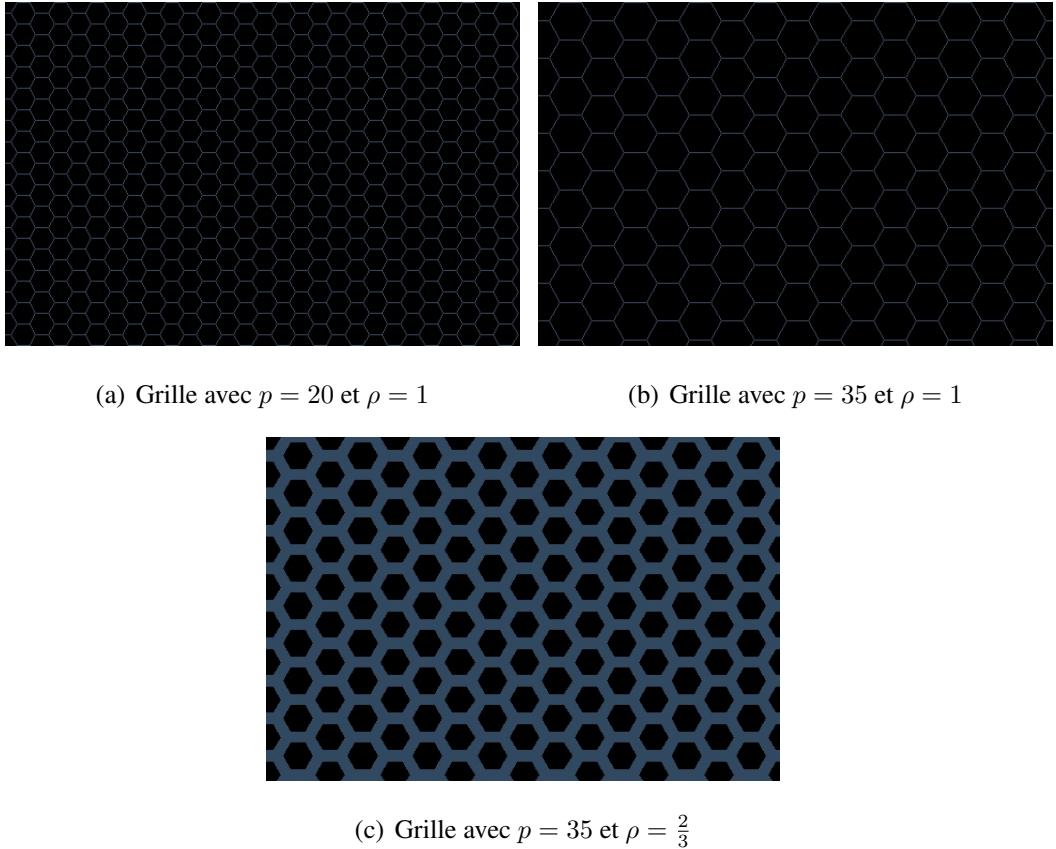


Figure 2: Exemple de grilles hexagonales selon différentes valeurs du pas de la grille p et du facteur d’homothétie ρ

Ainsi, rajouter une marge garantit une distance minimale entre le marqueur de la cellule C et les marqueurs des cellules voisines puisque tous les potentiels marqueurs seront tronqués si certains de leurs pixels se trouvent sur cette marge.

2.3 Sélection des marqueurs

L’une des forces de la méthode des Waterpixels est qu’elle assure la sélection d’un seul marqueur par cellule ce qui nous assure d’avoir un nombre de superpixels fixe (égal au nombre de cellules régulières définies).

De plus, afin d’assurer leurs connectivité et leurs régularité, nous prendrons pour marqueur de chaque cellule, la plus grande composante connexe obtenue en calculant le minimum du

gradient de cette cellule. La figure 3 montre pour une image donnée, les résultats de la sélection des marqueurs à partir des minimas locaux des cellules.

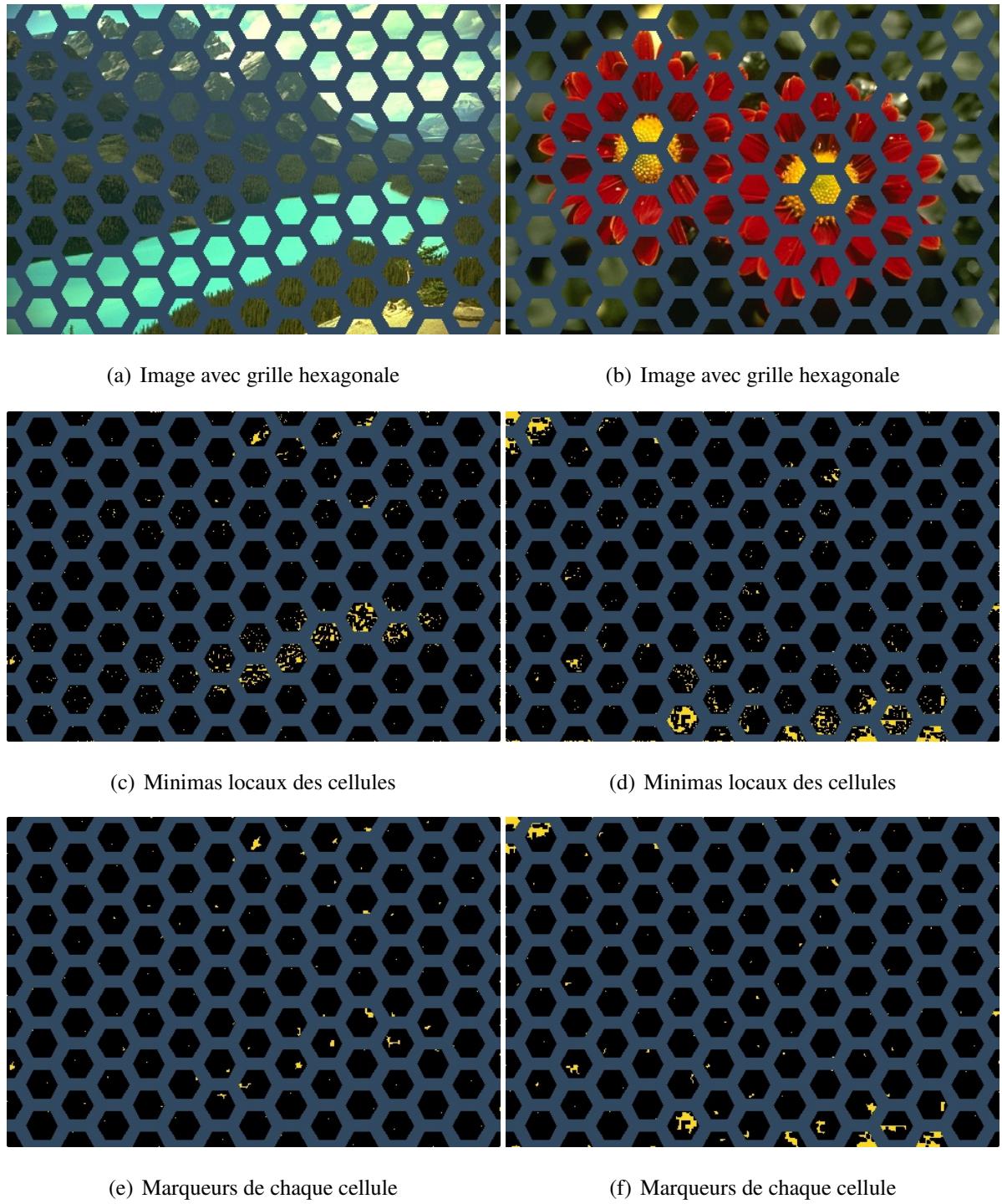


Figure 3: Quelques résultats de la sélection de marqueurs avec le pas de la grille $p = 40$ et un lissage gaussien d'écart-type $\sigma = 1$

2.4 Régularisation spatiale du gradient

L'étape de régularisation spatiale du gradient sert à équilibrer entre la régularité des superpixels et leurs adhérence aux contours. Le gradient résultant g_{reg} est défini comme suit:

$$g_{reg} = g + \frac{2}{p} k \min_{m \in M} d_Q(x, m) \quad (1)$$

Où g est le gradient initialement calculé et M est l'ensemble des marqueurs choisis.

d_Q est une fonction de distance qui affecte à chaque pixel x d'une image, la distance entre ce pixel et le marqueur m le plus proche de lui selon une métrique donnée. A l'issu de ce calcul, on aura généré une carte distance formant des *Tessellations de Voronoï* que l'on peut définir comme étant un partitionnement de l'image en un ensemble de régions adjacentes appelées «*cellules de Voronoï*» où chaque cellule regroupe des pixels voisins plus proches du marqueur de cette cellule que de tous les marqueurs restants.

Quant au facteur de régularisation k , il régule l'importance donnée aux parties du gradient correspondantes aux contours dessinés par les *Tessellations de Voronoï* lors de l'application de l'algorithme de la ligne de partage des eaux.

De ce fait, si k se rapproche de 0, aucune régularisation n'est appliquée. Si au contraire, sa valeur est assez grande, les formes des superpixels obtenus se rapprocheront de plus en plus de celles des *cellules de Voronoï* définies.

Il est à noter qu'une normalisation par le pas de la grille p est nécessaire pour rendre la régularisation invariante au nombre de superpixels choisis.

La figure 4 illustre les résultats de la régularisation spatiale du gradient.

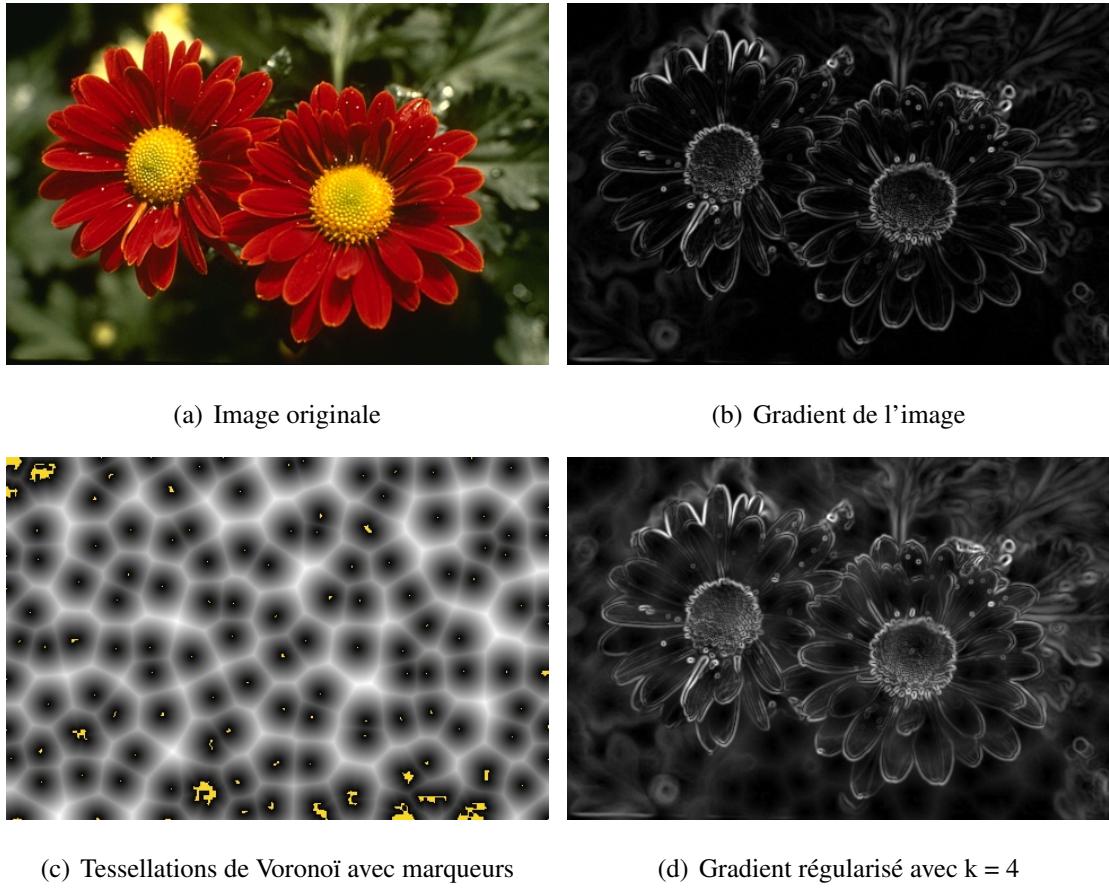


Figure 4: Résultat de la régularisation spatiale du gradient en utilisant l’opérateur de Sobel avec un lissage gaussien d’écart-type $\sigma = 1$, le pas de la grille $p = 40$ et un facteur de régularisation $k = 4$

2.5 Algorithme de la ligne de partage des eaux

Le principe de cet algorithme est de considérer notre image gradient comme étant un relief topographique que l’on inonde à partir de différentes régions appelées «*sources d’eau*» [7].

Au cours de l’immersion de la surface dans l’eau, dès que des eaux de différentes sources se rencontrent, on empêche leur fusion en les séparant par une barrière que l’on nommera «*ligne de partage des eaux*».

Une fois que la surface est complètement submergée dans l’eau, on aura partitionné notre image en deux ensembles: un ensemble de bassins versants qui correspondent aux régions homogènes de l’image et un ensemble de lignes de partage des eaux qui correspondent aux

contours de ces régions.

La dernière étape de la méthode des Waterpixels requiert donc que l'on applique l'algorithme de la ligne de partage des eaux en spécifiant le gradient régularisé comme étant le relief à inonder et l'ensemble de marqueurs sélectionnés comme étant les sources de l'inondation. La figure 5 illustre les résultats finaux que l'on peut obtenir par la méthode des Waterpixels.

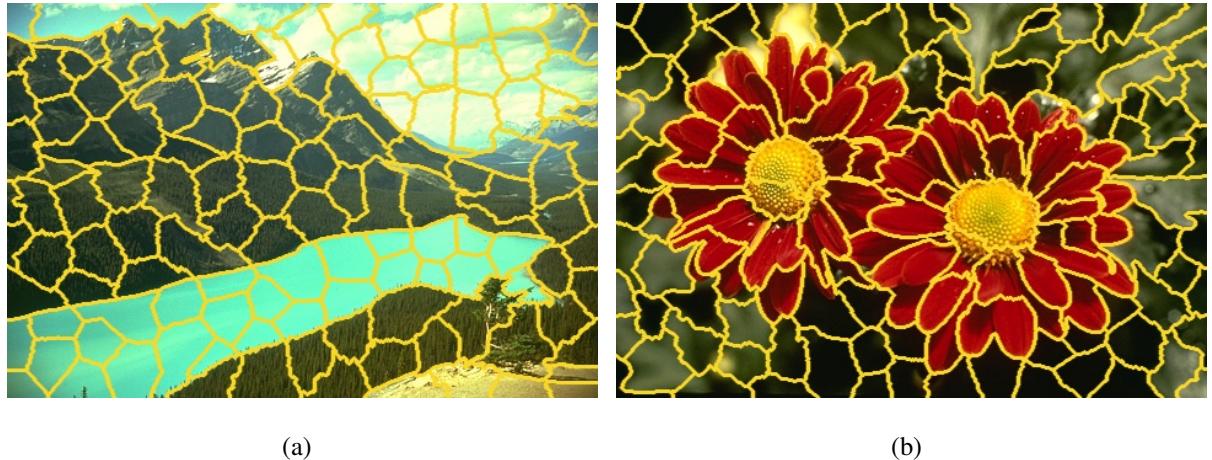


Figure 5: Waterpixels obtenus en utilisant l'opérateur de Sobel avec un lissage gaussien d'écart-type $\sigma = 1$, un pas de grille $p = 40$ et un facteur de régularisation $k = 4$

3 Implémentation et choix des paramètres

Les performances de la méthode des Waterpixels dépend de plusieurs paramètres, certains sont liés à l'image considérée (sa résolution, forte présence de bruit, etc...) tandis que d'autres sont relatifs à la nature de la méthode (gradient utilisé, distance qui caractérise la carte distance, taille de la grille hexagonale, etc..).

Avec la perspective d'améliorer la méthode des Waterpixels, on explicitera dans cette section tous les choix d'implémentation effectués et on étudiera l'influence des différents paramètres de la méthode sur quelques images test.

3.1 Calcul du gradient

On compare dans cette partie les résultats de la méthode des Waterpixels selon: l'opérateur de Sobel et le gradient morphologique.

L'opérateur de Sobel : comme l'on utilise deux filtres de tailles 3×3 pour estimer les changements d'intensité dans les deux directions : verticale et horizontale, on définit la norme du gradient G d'une image I comme illustré dans la figure 6.

$$G_x = \begin{array}{|c|c|c|} \hline 1 & 0 & -1 \\ \hline 2 & 0 & -2 \\ \hline 1 & 0 & -1 \\ \hline \end{array} * I \quad , \quad G_y = \begin{array}{|c|c|c|} \hline 1 & 2 & 1 \\ \hline 0 & 0 & 0 \\ \hline -1 & -2 & -1 \\ \hline \end{array} * I$$

$$G = \sqrt{G_x^2 + G_y^2}$$

Figure 6: Formulation de la norme du gradient selon l'opérateur de Sobel

Où G_x et G_y sont obtenus par convolution de l'image I respectivement avec le filtre horizontal et le filtre vertical de Sobel.

Le filtre de Sobel est connu pour être considérablement sensible au bruit. C'est pourquoi, on cherche d'abord à atténuer le bruit en pré-traitant nos images à l'aide d'un filtre gaussien de paramètre σ qui contrôle la netteté de l'image résultante.

Le gradient morphologique : fait ressortir les contours des objets en appliquant sur une image I , la différence entre deux opérations duales de la morphologie mathématique: la dilatation et l'érosion. Tandis que l'érosion cherche à rétrécir les objets de l'image, la dilatation vise à les agrandir.

Les deux opérations font appel à la notion d'élément structurant H qui fait office de fenêtre glissante de taille fixe que l'on déplace sur notre image comme lors d'une convolution. Les pixels de H mis à 1 définissent sa forme (carré, croix, cercle...) et permettent de déterminer le voisinage de pixels de l'image à considérer lors d'une opération morphologique.

La dilatation d'une image en niveau de gris consistera donc à assigner à chacun de ses pixels x , la valeur maximale des pixels du voisinage défini par l'élément structurant centré sur x . Pareillement, l'érosion est définie comme étant le minimum entre les valeurs des pixels du voisinage défini par l'élément structurant H .

Nous testons maintenant notre implémentation avec différentes valeurs de σ pour le lissage gaussien précédant l'opérateur de Sobel et différentes formes de l'élément structurant pour le gradient morphologique.

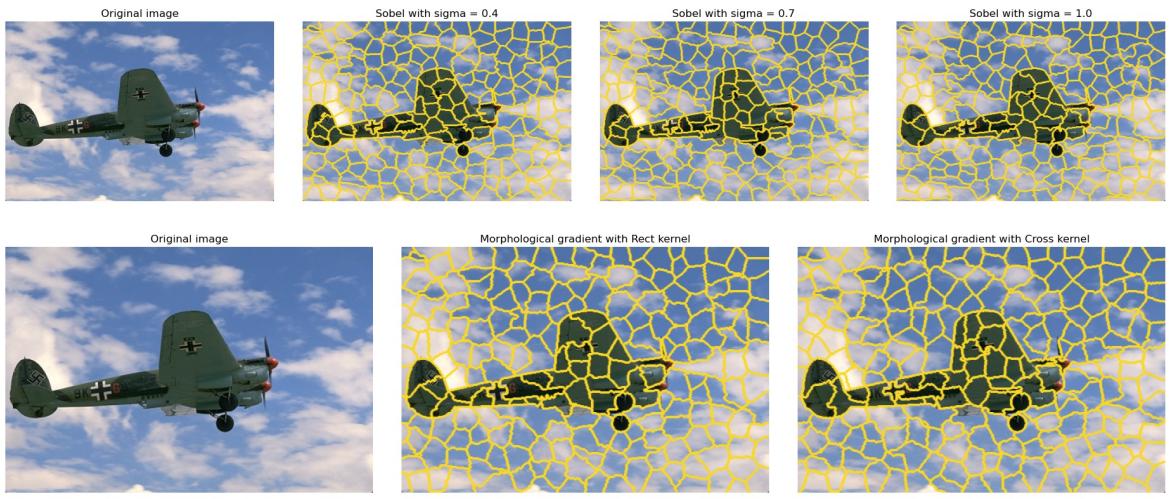


Figure 7: Waterpixels obtenus en utilisant l'opérateur de Sobel avec un lissage gaussien d'écart-type σ et le gradient morphologique en fixant le pas de grille p à 30 et le facteur de régularisation k à 4

On constate à travers l'exemple de la figure 7 qu'en augmentant le paramètre du lissage gaussien σ , moins de détails ne sont considérés et les superpixels ont des formes beaucoup plus régulières sur les zones texturées.

Il est à noter que le filtre gaussien est un filtre passe bas, il atténue donc les hautes fréquences sans pouvoir dissocier les pixels bruités des contours. C'est pourquoi, plus on augmente la valeur de σ , plus on risque d'éliminer de bons contours délimitant réellement des objets de l'image (c'est ce que l'on observe pour la croix à l'arrière de l'avion).

D'autre part, des superpixels beaucoup plus homogènes sont formés et beaucoup plus de faux contours sont ignorés, comme on peut le voir dans la figure 8, sur les roues et le devant de l'avion :

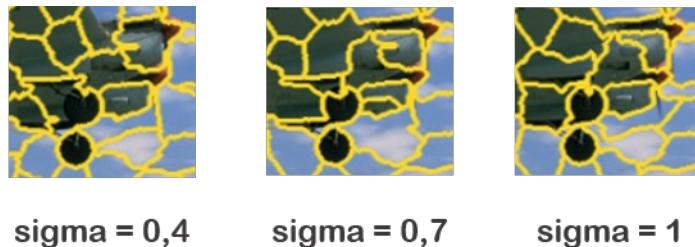


Figure 8: Zoom sur les Waterpixels obtenus en utilisant l’opérateur de Sobel avec le pas de grille $p = 30$ et le facteur de régularisation $k = 4$

On remarque également que le gradient morphologique dessine un peu mieux les formes similaires à la forme de son élément structurant. Par exemple, le gradient morphologique avec un élément structurant en croix est celui qui dessine le mieux les contours de la croix sur l’avion.

Globalement, même si utiliser l’opérateur de Sobel requiert que l’on fixe le taux de netteté souhaité, les contours qu’il dessine sont plus précis que ceux du gradient morphologique et les superpixels produits sont beaucoup plus homogènes (on le distingue clairement sur la forme des nuages des deux dernières images avec un $\sigma = 1$ et un cross kernel). C’est pourquoi dans la suite du présent document, on utilisera l’opérateur de Sobel pour calculer le gradient d’une image.

3.2 Définition de la carte distance

L’idée derrière le calcul de la carte distance est d’approximer les distances globales entre chaque pixel de l’image et les marqueurs sélectionnés en propageant les distances locales (distances entre des pixels voisins).

Les distances locales sont représentées par des masques que l’on subdivise en deux parties. Dans un premier temps, on fait parcourir le premier demi-masque obtenu (le demi masque-haut) sur toute l’image de haut en bas et de gauche à droite puis on fait parcourir le deuxième demi-masque de bas en haut et de droite à gauche.

La nouvelle valeur d’un pixel central est obtenue en retenant la valeur minimale de la somme de chaque distance locale du demi-masque et de la valeur du pixel correspondant dans

l'image.

On considère donc une image binaire de même taille que l'image d'entrée où seuls les marqueurs sont étiquetés à 0. Nous définissons ensuite un masque de tailles 5×5 en prenant les distances locales proposées par Borgefors [8] comme illustré sur la figure 9 pour minimiser les erreurs d'approximation de la distance euclidienne.

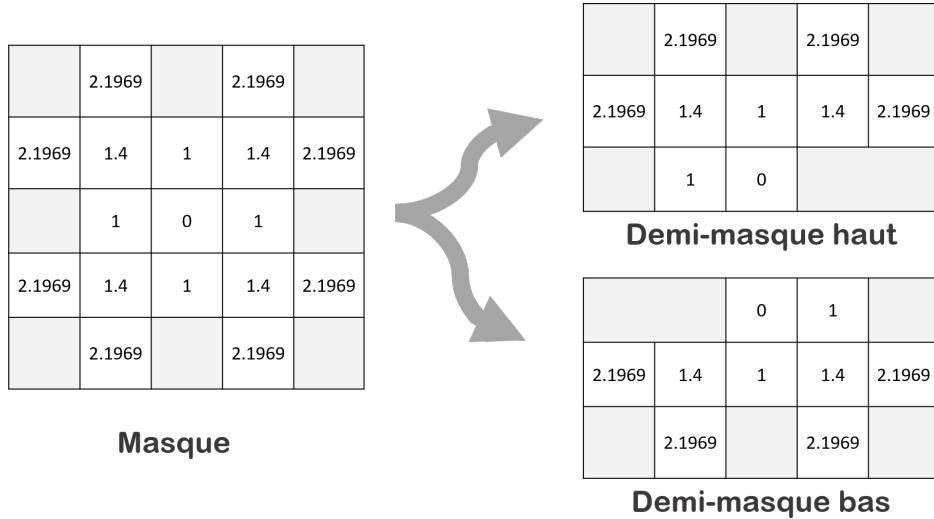


Figure 9: Illustration du masque des distances locales

Dans un second temps et afin d'aboutir à un calcul plus précis et rapide de la distance euclidienne, on a finalement opté pour l'application de l'algorithme de transformation de distance euclidienne proposé par Felzenszwalb & al[9].

3.3 Taille de la grille hexagonale

La grille hexagonale dépend de deux paramètres : le pas de grille p et le facteur d'homothétie ρ que l'on fixe à $\frac{2}{3}$ tout au long de notre étude.

Le pas de la grille hexagonale contrôle le nombre et la taille des superpixels produits et affecte en conséquence l'adhérence aux contours des objets ainsi que le temps d'exécution de la méthode. Cela dit, fixer ce paramètre dépend entièrement des détails que l'on souhaite capturer: selon que l'on veuille dessiner les contours de petits ou grands objets.

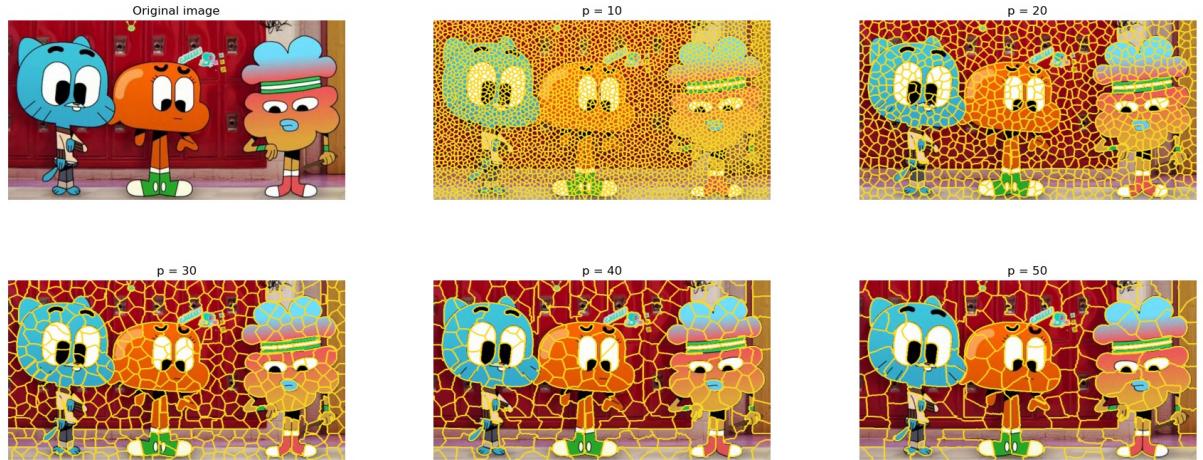


Figure 10: Waterpixels obtenus en variant le pas de grille p avec le facteur de régularisation $k = 10$ et un écart-type $\sigma = 0.3$ pour le lissage gaussien

On observe sur la figure 10 que lorsque l'on fixe une petite valeur du pas de la grille, on a tendance à mieux capturer les petits détails (lorsque $p = 20$, on arrive à dessiner le bras du poisson orange, le médaillon sur le casier...). Plus on augmente la valeur du pas p , moins l'on a de superpixels, qui eux sont de plus grandes tailles et dessinent les contours de plus grands objets (ce qui est le cas des vêtements et des yeux des personnages lorsque $p = 50$).

3.4 Facteur de régularisation k

Si l'on reprend l'image précédente en faisant varier le facteur de régularisation k , on obtient les Waterpixels de la figure 11.



Figure 11: Waterpixels obtenus avec $p = 40$ et faisant varier le facteur de régularisation k et un écart-type $\sigma = 0.3$ pour le lissage gaussien

On voit bien que plus on augmente la valeur du facteur de régularisation k , plus on génère des superpixels de formes et de tailles régulières (qui se rapprochent des Tesselations de Voronoï associées) au dépend de l’adhérence aux contours dans certaines régions. Ceci dit, en faisant de multiples tests, on a déduit qu’il n’est pas possible de fixer une valeur idéale de k pour toute image puisque chaque image contient divers objets de tailles et de formes différentes.

4 Évaluation de la méthode

Trois critères sont pris en compte pour jauger la pertinence de notre implementation: l’adhérence aux contours des objets, la régularité des superpixels et l’efficacité en temps de calculs. Pour chacun d’eux, nous définissons et calculons un certain nombre de métriques et on comparera les résultats obtenus par les Waterpixels avec ceux obtenus par une version adaptative de l’algorithme SLIC nommée SLIC0 [10] où le paramètre de régularisation est automatiquement adapté pour chaque superpixels.

Les tests sont effectués sur un Intel(R) Core(TM) i7-8550U CPU @ 1.8 GHZ de 4 coeurs sur la base de 100 images provenant de deux bases d’images différentes:

Berkeley Segmentation Database (BSDS500) [11] : est un ensemble de 500 images naturelles de tailles égales (321 x 481 ou 481 x 321) dont chaque image dispose d’au moins 5

vérité terrain. Quelques exemples d’images avec leurs vérité terrain sont présentés dans la figure 12



Figure 12: Exemple d’images de la base BSDS500 avec leurs vérité terrain

Stanford Background Dataset (SBD) [12]: comprend 715 images naturelles de différentes tailles (majoritairement 320 x 240) et qualités, accompagnées d’images étiquetant chaque région, comme le montre la figure 13.



Figure 13: Exemple d’images de la base SBD avec leur vérité terrain

4.1 Adhérence aux contours des objets

Boundary recall est une métrique communément utilisée pour quantifier l’adhérence des superpixels aux contours des objets. Cette dernière évalue le pourcentage des contours de

superpixels c qui coïncident avec les contours v des images de référence suivant une distance minimale d_{min} :

$$BR = \frac{|\{v \in GT, d(v, c) < d_{min}\}|}{|GT|} \quad (2)$$

Où la distance minimale d_{min} est fixée à trois et d est la distance de Manhattan.

Cela dit, il a été rapporté par Pavel & all[13] que lorsque l'on a des superpixels aux contours irréguliers, il est plus probable d'obtenir de plus grandes valeurs de BR puisque les contours fluctueux sont beaucoup plus présents et ont plus de chance de correspondre aux contours de l'image référence.

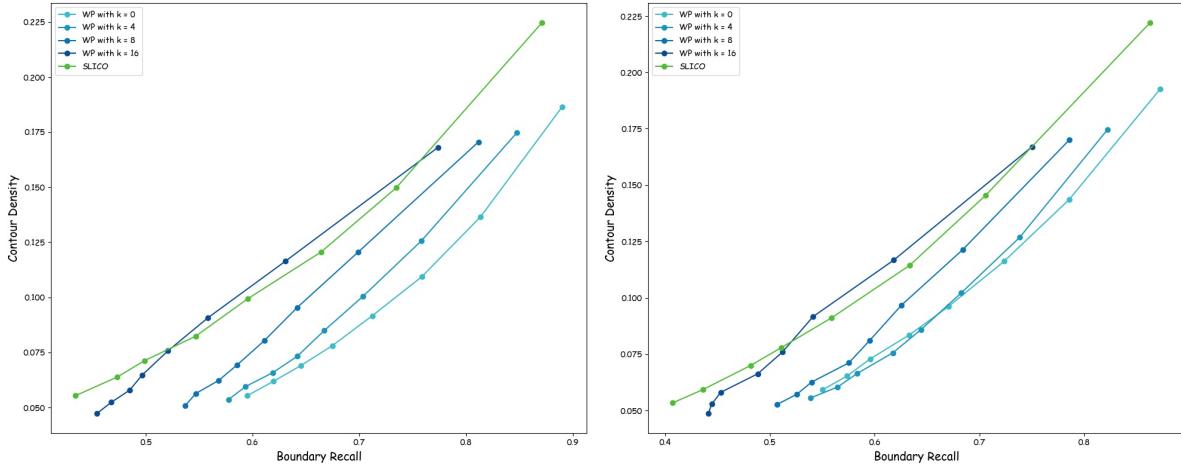
Afin d'aboutir à une évaluation plus correcte, Vaïa & all [1] estiment que les superpixels qui adhèrent le mieux aux contours des objets doivent minimiser le nombre de contours dessinés tout en maximisant leur correspondance aux contours des images références.

Pour mesurer le nombre de contours des superpixels S_c présents dans l'image, on calcule la densité des contours que l'on note CD :

$$CD = \frac{|S_c|}{|P|} \quad (3)$$

$|S_c|$ étant le nombre de pixels contours des superpixels et $|P|$ étant le nombre de pixels dans l'image.

De ce fait, des Waterpixels qui ont une bonne adhérence aux contours des objets doivent maximiser BR tout en maintenant une valeur minimale de CD. La figure 14 illustre l'évaluation de cet équilibre sur les deux bases d'images pour différents pas de la grille p tel que $p \in \{t * 5 \mid 2 \leq t < 10, t \in \mathbb{N}\}$.



(a) Évaluation sur 100 images de la base BSDS

(b) Évaluation sur 100 images de la base SBD

Figure 14: Résultats de l'évaluation moyenne de l'adhérence des superpixels aux contours des objets avec l'écart type $\sigma = 1$
et différents pas de grille p

Discussion des résultats obtenus:

- On observe pour les deux jeux de données que plus le facteur de régularisation k des Waterpixels est petit, plus le nombre de contours nécessaires pour coller aux contours des objets est minimisé, ce qui est tout à fait logique puisque plus on augmente k plus les Waterpixels tendent vers *les Tesselations de Voronoï* et adhèrent de moins en moins aux contours des objets
- Même si idéalement, on souhaiterai maximiser le boundary recall avec une densité de contours minimale, on peut dire que lorsque $k = 4$, les Waterpixels donnent des résultats plutôt satisfaisants puisque pour les deux bases de données on arrive à couvrir plus de 80% des contours réels des images avec une densité de contours ne dépassant pas les 20%
- Il reste indéniable que les Waterpixels offrent une meilleure adhérence aux contours des objets tout en formant des superpixels avec moins de contours que SLIC0 et cela reste valable en augmentant le facteur de régularisation jusqu'à un certain seuil (lorsque $k = 16$, SLIC0 surpassé les Waterpixels) ou même en omettant l'étape de régularisation

4.2 Régularité des superpixels

On évalue la régularité de la décomposition S d'une image I à l'aide de la mesure de régularité globale (GR) proposée par Giraud & all[14]. Celle-ci prend en compte deux aspects: la régularité des formes des superpixels que l'on calcule à travers le critère de régularité des formes (SRC) et leur consistance que l'on mesure grâce au smooth matching factor (SMF) avec:

$$GR = SRC \times SMF \quad (4)$$

Shape regularity criteria (SRC) : ce critère est défini de sorte à encourager l'obtention de superpixels de formes convexes et de contours lisses tout en veillant à ce que les pixels soient répartis de façon équilibrée:

$$SRC = \sum_{S_k} \frac{|S_k|}{|I|} CR(S_k) \sqrt{V_{xy}(S_k)} \quad (5)$$

En calculant le critère de régularité CR d'un superpixel S_k , on considère la convexité de sa forme puisque l'on évalue la régularité de ses contours ainsi que les chevauchements avec son enveloppe convexe:

$$CR(S_k) = \frac{CC(H_k)}{CC(S_k)} \quad (6)$$

Où H_k est l'enveloppe convexe du superpixel et CC est le rapport entre le carré du périmètre $P(F)$ et l'aire d'une forme F :

$$CC(F) = \frac{|P(F)|^2}{|F|} \quad (7)$$

Afin d'éviter de donner un grand score à des formes convexes telles que les lignes ou les ellipses, on pénalise les dissimilarités dans la répartition des pixels de la forme en définissant le ratio V_{xy} entre la variance minimale et maximale des positions x et y des pixels de S_k :

$$V_{xy} = \frac{\min(\sigma_x, \sigma_y)}{\max(\sigma_x, \sigma_y)} \quad (8)$$

σ_x et σ_y étant les écarts types des positions des pixels.

Ainsi, plus CR se rapproche de 1, plus la forme étudiée est convexe et plus on a une distribution déséquilibrée des pixels, plus SRC diminue.

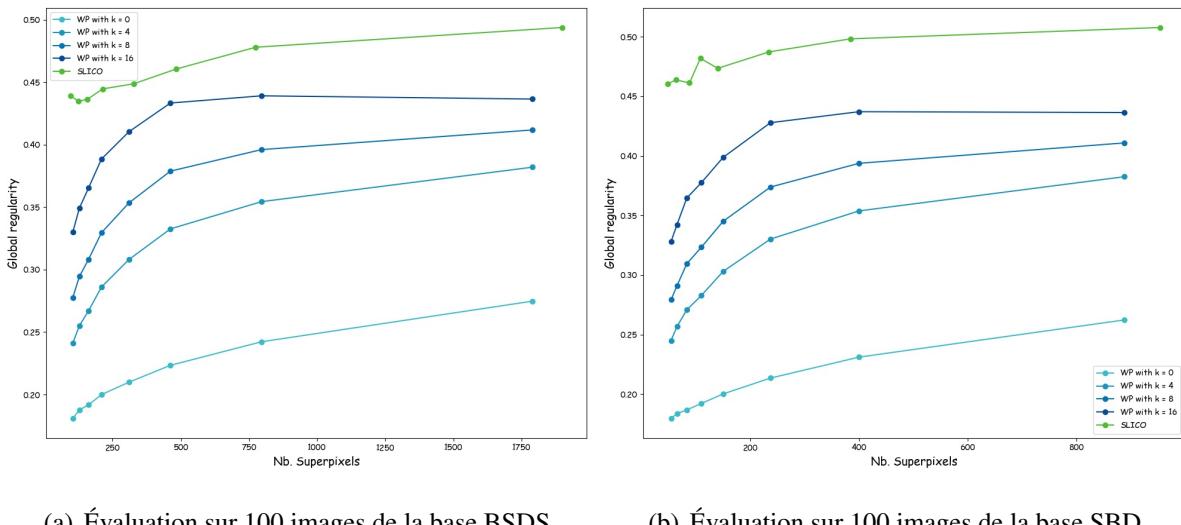
Smooth matching factor (SMF) : mesure la similarité entre les superpixels en terme de taille et de forme en comparant la distribution spatiale des superpixels centrés S_k^* avec celle du superpixel centré moyen S^* :

$$SMF = 1 - \sum_{S_k} \frac{|S_k|}{|I|} \times d \left(\frac{S^*}{|S^*|}, \frac{S_k^*}{|S_k^*|} \right) / 2 \quad (9)$$

Où d est la distance de Manhattan et S^* est défini comme suit:

$$S^* = \frac{1}{|D|} \sum_{S_k} S_k^* \quad (10)$$

Nous présentons dans la figure 15, les résultats de l'évaluation de la régularité globale des superpixels sur les deux bases d'images pour différents pas de la grille p tel que $p \in \{t * 5 \mid 2 \leq t < 10, t \in \mathbb{N}\}$.



(a) Évaluation sur 100 images de la base BSDS

(b) Évaluation sur 100 images de la base SBD

Figure 15: Résultats de l'évaluation moyenne de la régularité globale en fonction du nombre de superpixels avec l'écart type $\sigma = 1$ et différents pas de grille p

Discussion des résultats obtenus:

- Comme on aurait pu l'attendre, lorsque l'on ignore l'étape de régularisation du gradient, on obtient des Waterpixels très irréguliers alors qu'on observe une nette amélioration du taux de régularité dès lors que l'on augmente la valeur du facteur de régularisation k , ce qui montre l'importance de l'étape de régularisation.
- On constate également que la version améliorée de SLIC surpassé la méthode des Waterpixels en terme de régularité, cela étant dû au fait que la valeur du paramètre de régularisation de SLIC0 est adaptée pour chaque superpixel formé afin de renforcer la régularité (au détriment de l'adhérence au contours selon les constatations faites à partir des courbes de la figure 14).

4.3 Efficacité de l'implémentation

Nous avons remarqué durant nos tests que deux opérations étaient particulièrement coûteuses en temps de calcul: *la sélection des marqueurs* qui nécessite que l'on trouve pour chaque cellule, la plus grande composante connexe de gradient minimal et *la génération de la carte distance* étant donné que pour chaque pixel de l'image, on doit calculer sa distance avec tous les marqueurs dont on dispose et lui affecter la distance minimale.

Afin d'optimiser ces calculs, on a privilégié l'utilisation de méthodes optimisées fournies par Numpy et d'autres librairies telle que OpenCV qui nous a permis de considérablement accélérer la génération de la carte distance grâce à la méthode « *distanceTransform()* » qui implémente l'algorithme de calcul de la carte distance sélectionné [9].

Ainsi, nous avons comparé sur 100 images de la base BSDS et SBD, le temps d'exécution moyen de notre implémentation en prenant toujours différentes valeurs du pas de la grille p tel que $p \in \{t * 5 \mid 2 \leq t < 10, t \in \mathbb{N}\}$. Les résultats sont présentés dans la figure 16.

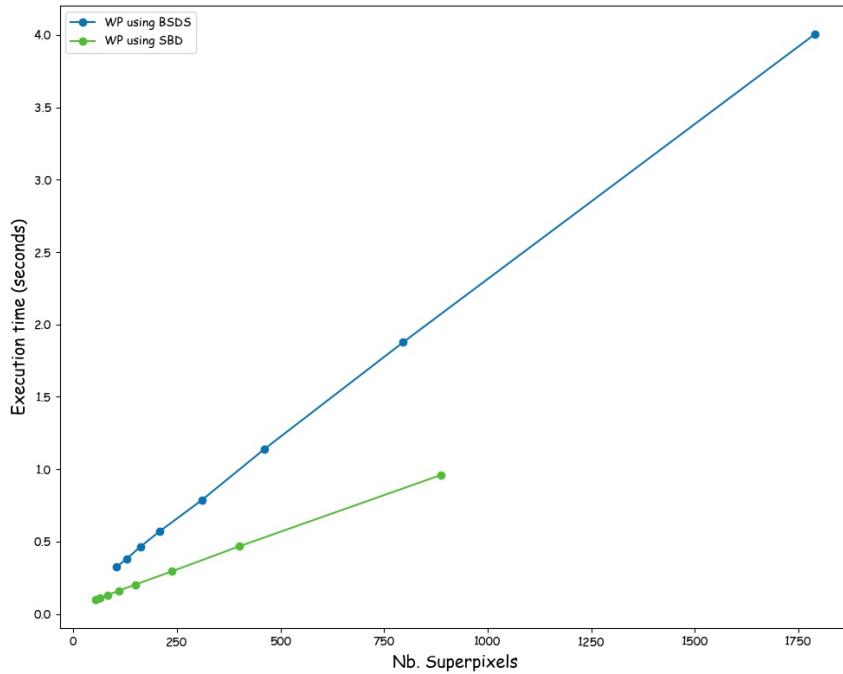


Figure 16: Résultats de l'évaluation moyenne du temps d'exécution de notre implementation

En moyenne, on a un temps de calcul qui croît de façon linéaire en fonction du nombre de superpixels. Cependant, il est à noter que le temps d'exécution dépend aussi de la taille des images: les images de la base BSDS étant deux fois plus grandes que celles de la base SBD, l'écart entre les deux courbes est également doublé.

5 Application de bureau

Dans l'optique de permettre à tout utilisateur de facilement tester notre implementation de la méthode des Waterpixels, on a réalisé une application de bureau codée en Python via l'API Qt5 et présentée à travers une interface intuitive:

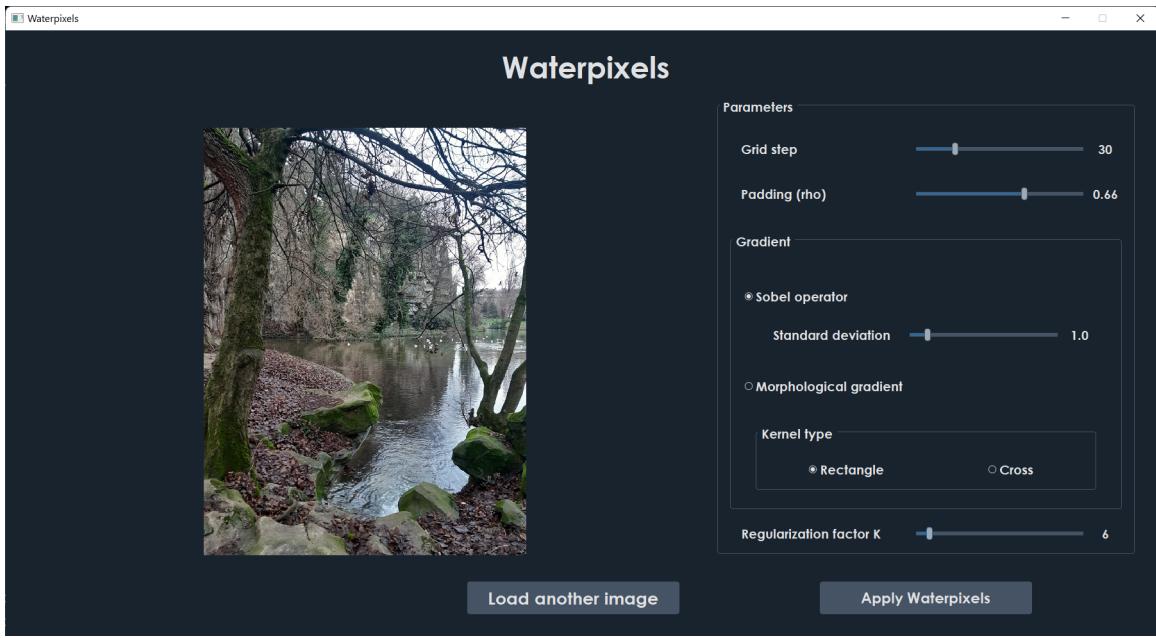


Figure 17: Interface de chargement de l'image et de sélection des paramètres

Comme on peut le voir sur la figure 17, l'utilisateur a la possibilité de charger une image à partir de n'importe quel répertoire et d'y appliquer la méthode des Waterpixels en fixant les paramètres souhaités. Une fois que la génération des Waterpixels finit, on peut visualiser les résultats de chaque étape à travers l'interface de la figure 18.

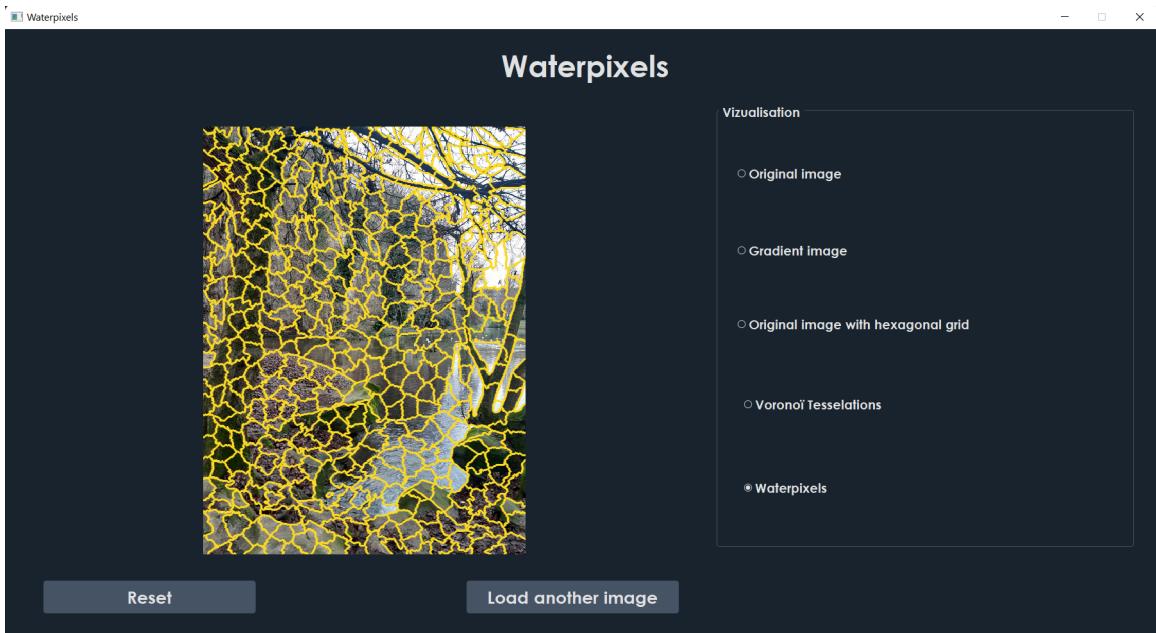


Figure 18: Interface de visualisation des Waterpixels

Conclusion

Suites aux expérimentations et aux tests effectués, on peut conclure que notre implémentation de la méthode des Waterpixels se révèle être assez robuste puisqu'elle parvient à satisfaire deux critères majeurs: l'adhérence aux contours des objets et la régularité globale des superpixels. De même que les temps d'exécution sont assez satisfaisants même sur des images moyennement grandes.

D'autant plus que d'autres modifications peuvent être apportées afin de l'améliorer. On pourrait par exemple, comme pour le cas de SLIC0, essayer d'adapter le facteur de régularisation k pour chaque superpixel ou encore de faire varier la forme de la grille régulière. Une autre perspective d'amélioration serait de réguler le nombre de superpixels selon l'homogénéité des régions, ce qui donnerai moins de superpixels dans les régions homogènes et plus de superpixels dans celles qui le sont moins.

BIBLIOGRAPHIE

- [1] V. Machairas et al. “Waterpixels”. In: *Image Processing, IEEE Transactions on* 24 (Nov. 2015), pp. 3707–3716. DOI: 10.1109/TIP.2015.2451011.
- [2] X. Ren and J. Malik. “Learning a classification model for segmentation”. In: *ICCV 2003: 9th International Conference on Computer Vision* 1 (Nov. 2003), pp. 10–17. DOI: 10.1109/ICCV.2003.1238308.
- [3] D. Stutz, A. Hermans, and B. Leibe. “Superpixels: An Evaluation of the State-of-the-Art”. In: *Computer Vision and Image Understanding* 166 (Jan. 2018), pp. 1–27. DOI: 10.1016/j.cviu.2017.03.007.
- [4] P. Neubert and P. Protzel. “Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms”. In: (Aug. 2014), pp. 996–1001. DOI: 10.1109/ICPR.2014.181.
- [5] A. Vedaldi and S. Soatto. “Quick Shift and Kernel Methods for Mode Seeking”. In: *Computer Vision – ECCV 2008*. Vol. 5305. Series Title: Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2008, pp. 705–718. DOI: 10.1007/978-3-540-88693-8_52.
- [6] M. Van den Bergh et al. “SEEDS: Superpixels Extracted via Energy-Driven Sampling”. In: *International Journal of Computer Vision* 111 (Oct. 2012). DOI: 10.1007/978-3-642-33786-4_2.
- [7] S. Beucher and F. Meyer. “The Morphological Approach to Segmentation: The Watershed Transformation”. In: *Mathematical Morphology in Image Processing*. CRC Press, 1993, pp. 433–481. DOI: 10.1201/9781482277234-12.
- [8] Borgefors G. “Distance transformations in digital images”. In: *Computer Vision, Graphics, and Image Processing* 34.3 (1986), pp. 344–371. ISSN: 0734-189X. DOI: [https://doi.org/10.1016/S0734-189X\(86\)80047-0](https://doi.org/10.1016/S0734-189X(86)80047-0).
- [9] Felzenszwalb P. and Huttenlocher D. “Distance Transforms of Sampled Functions”. In: *Theory Comput.* 8 (2012), pp. 415–428.

- [10] R. Achanta et al. “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34 (2012), pp. 2274–2282. DOI: 10.1109/TPAMI.2012.120.
- [11] P. Arbelaez, C. Fowlkes, and D. Martin. *The Berkeley Segmentation Dataset and Benchmark*. June 2007. (Visited on 01/05/2022).
- [12] S. Gould, R. Fulton, and D. Koller. “Decomposing a Scene into Geometric and Semantically Consistent Regions”. In: *Proceedings of the IEEE International Conference on Computer Vision* (Nov. 2009), pp. 1–8. DOI: 10.1109/ICCV.2009.5459211.
- [13] P. Kalinin and A. Sirota. “A graph based approach to hierarchical image over-segmentation”. In: *Computer Vision and Image Understanding* 130 (Jan. 2014). DOI: 10.1016/j.cviu.2014.09.007.
- [14] R. Giraud, V. Ta, and N. Papadakis. “Evaluation framework of superpixel methods with a global regularity measure”. In: *Journal of Electronic Imaging* 26 (Nov. 2017). DOI: 10.1117/1.JEI.26.6.061603.