

République Algérienne Démocratique et Populaire
Ministère de L'Enseignement Supérieur et de la A Recherche Scientifique

Université Mouloud Mammeri De Tizi-Ouzou



Faculté De Génie Electrique Et D'informatique
DEPARTEMENT D'AUTOMATIQUE

**Mémoire de Fin d'Etudes
de MASTER ACADEMIQUE**
Spécialités : Automatique et Systèmes,
Automatique et informatique industriel

Présenté par
Amar CHALAL
Kamel AIT HAMICHE

Mémoire dirigé par Mme Sadia ALKAMA

Thème

Segmentation d'images en utilisant la superpixelisation

Mémoire soutenu publiquement le 24/09/ 2018 devant le jury composé de :

M Kamal HAMMOUCHE
Professeur, UMMTO, Président

Mme Sadia ALKAMA
MCA, UMMTO, Rapporteur

Mme Farida DORBANE
MAA, UMMTO, Examineur

Mme Ouardia CHILALI
MCB, UMMTO, Examineur

Remerciements

Nous tenons à exprimer nos profonds remerciements et notre vive reconnaissance et considération à notre promotrice. Sadia ALKAMA, qui a su, à sa façon, nous conseiller et nous orienter tout au long de la réalisation de ce travail.

Nos remerciements particuliers sont adressés au cadre pédagogique de nous avoir consacré de leur temps et nous avoir apporté de l'aide par leurs précieux conseils.

Que toute personne ayant participé de près ou de loin dans l'élaboration de ce travail, trouve ici l'expression de notre vive reconnaissance.

Je dédie ce modeste travail

A mes parents, aucun hommage ne pourrait être à la hauteur de l'amour dont ils ne cessent de me combler. Que dieu leur procure bonne santé et longue vie.

A mes chers frères et sœurs pour leur affection et leur encouragement qui ont toujours été pour moi des plus précieux.

Que ce travail soit pour vous le gage de notre profond amour

A tous nos amis

A tous ceux qui m'ont aidé

A tous ceux qui m'aime

INTRODUCTION GENERALE

Chapitre I : Segmentation d'image

I.1. Introduction	3
I.2. Définition de la segmentation	3
I.3. Approches de la segmentation d'images	4
I.3.1. Approche contour	5
I.3.2. Approche région	6
I.3.2.1. Segmentation par division de région	6
I.3.2.2. Segmentation par fusion	7
I.3.2.3. Segmentation par division et fusion de région (split and merge)	8
I.3.3. Classification	8
I.3.3.1. Méthodes supervisées	9
I.3.3.2. Méthodes non-supervisée	9
I.4. Quelques méthodes de classification.....	10
I.4.1. Méthode k plus proches voisins (K-PPV)	10
I.4.2. Méthodes k-médoids	11
I.4.3. Méthode des k-means	12
I.5. Conclusion	16

Chapitre II : Superpexilisation

II.1. Introduction	17
II.2. Définition de la superpexilisation	17
II.3. Méthodes de décomposition en superpixels.....	19
II.3.1. Méthodes de graphes	19
II.3.2. Méthodes hiérarchiques	20
II.3.3. Méthodes de montée de gradient.....	21
II.3.4. Comparaison des méthodes de décomposition en superpixels	21
II.4. Méthode SLIC	23
II.5 La méthode SCALP	27

II.5.1. Chemin linéaire	28
II.5.2. Distance couleur au superpixel	29
II.5.3. Adhérence à la carte contour	30
II.5.4. Algorithme SCALP.....	31
II.6. Conclusion	32

Chapitre III : Tests et résultats

III.1. Introduction	33
III.2. Préambule	33
III.3. Tests	35
III.4. Interprétation	45
III.5. Conclusion.....	46
CONCLUSION GENERALE	47
REFERANCES BIBLIOGRAPHIQUE	

Liste des figures

Chapitre I : segmentation d'image

Figure I.1 : Schéma d'un système de traitement d'images.....	3
Figure I.2 : techniques de segmentation	4
Figure I.3 : Exemple de détection de contour.....	5
Figure I.4 : Exemple de détection de contour en utilisant les opérateurs de Sobel et Prewitt	6
Figure I.5 : Exemple de segmentation par division de région.....	7
Figure I.6 : Exemple de segmentation par fusion.....	7
Figure I.7 : Exemple de division-fusion de région8
Figure I.8 : le principe de classification	8
Figure I.9 : Exemple de classification avec les K-PPV (K=5 et k=11)	11
Figure I.10 : Déroulement de l'algorithme de k-means	15

Chapitre II : Superpexilisation

Figure II.1 : Exemple d'image régulière et irrégulière.....	18
Figure II.2 : Illustration de la faible variance d'intensité dans les superpixels ..	19
Figure II.3: Graphe d'adjacence de pixels.....	20
Figure II.4 : Illustration du processus de décomposition de la méthode ETPS de type hiérarchique.....	21

Figure II.5 : Exemple de superpexilisation par l’algorithme SLIC, les contours des superpixels sont représentés en rouge	23
Figure II.6 : Illustration de la méthode de groupement itératif SLIC	25
Figure II.7 : Exemple de renforcement de la connexité	26
Figure II.8 : La méthode SCALP. Une carte de contours peut être utilisée pour améliorer le respect des contours de l’image.....	28
Figure II.9 : Illustration du chemin linéaire P_i^k entre un pixel i et un superpixel S_k de centre X_k	29
Figure II.10: Illustration de la première itération de la méthode SCALP.....	30

Chapitre III: Tests et résultat

Figure III.1: Exemple de superpexilisation d’une image.....	35
Figure III.2: résultat de la segmentation d’une image de taille 275x183 en faisant varier le Taux des pixels	37
Figure III.3: résultat de la segmentation d’une image de taille 1600x1024 en faisant varier le Taux des pixels	38
Figure III.4: résultat de la segmentation d’une image de taille 256x256 en faisant varier le Taux des pixels	39
Figure III.5: résultat de la segmentation d’une image de taille 1000x500 en faisant varier le Taux des pixels	40
Figure III.6: résultat de la segmentation d’une image de taille 481x321 en faisant varier le Taux des pixels	41
Figure III.7: résultat de la segmentation d’une image de taille 256x256 en faisant varier le Taux des pixels	42
Figure III.8: résultat de la segmentation d’une image de taille 481x321 en faisant varier le taux de superpexilisation	43

Avec la parole, l'image constitue l'un des moyens les plus importants qu'utilise l'homme pour communiquer avec autrui. C'est un moyen de communication universel dont la richesse du contenu permet aux êtres humains de tout âge et de toute culture de se comprendre. Chacun peut analyser l'image à sa manière, pour en dégager une impression et en extraire des informations précises. Le traitement d'images est l'ensemble des méthodes et techniques automatiques opérant sur celles-ci, dans le but d'en améliorer l'aspect visuel et d'en extraire des informations souhaitées.

Le traitement d'image est alors doté d'outils et de méthodes très puissantes telles que les mathématiques et l'informatique. Parmi les opérateurs de cette dernière, la segmentation est l'une des plus utilisées, son principe consiste à séparer les objets entre eux et du fond de l'image.

Notre travail s'inscrit dans ce domaine, il consiste à réduire le temps de calcul en réduisant le nombre de données à traiter. En effet, le temps de calcul durant la segmentation peut être important surtout lorsque l'image à segmenter est de grande dimension. Le nombre de données à traiter dans ce cas (c'est-à-dire le nombre de pixels) est très grand. Dans ce travail, nous proposons de réaliser une superpexilisation et réaliser la segmentation sur les images superpexilisées.

La segmentation sera effectuée en appliquant l'algorithme k-means sur, à la fois, les images superpexilisées et sur les images brutes afin de réaliser une comparaison des résultats.

Ce mémoire est divisé en trois chapitres :

Le premier chapitre présente les idées générales concernant la segmentation d'image et ces différentes approches.

Dans le deuxième chapitre, nous avons défini la notion de superpexilisation et nous avons développé deux techniques les plus connues, à savoir la méthode SLIC (simple linéaire itératif clustering) et La méthode SCALP (Superpixels with Contour Adhérence using Linear Path).

Le troisième et dernier chapitre, nous appliquons la méthode SLIC pour réaliser la décomposition en superpixels et l'algorithme k-means pour la segmentation. Nous présentons par la suite les résultats obtenus avant de réaliser une interprétation.

Nous terminons par une conclusion générale dans laquelle nous donnons quelques perspectives.

CHAPITRE I: **La segmentation** **d'images**

I.1. Introduction

Le traitement d'images est une analyse qui peut être défini comme l'ensemble des méthodes permettant de décrire quantitativement le contenu d'une image. Il est généralement décomposé en plusieurs étapes: acquisition, numérisation, prétraitement, segmentation et interprétation (voir figure I.1).

L'étape de la segmentation de l'image est une étape essentielle dans un système de traitement d'image car c'est à partir de l'image segmentée que les mesures sont effectuées sur les objets que contient l'image pour l'interprétation.

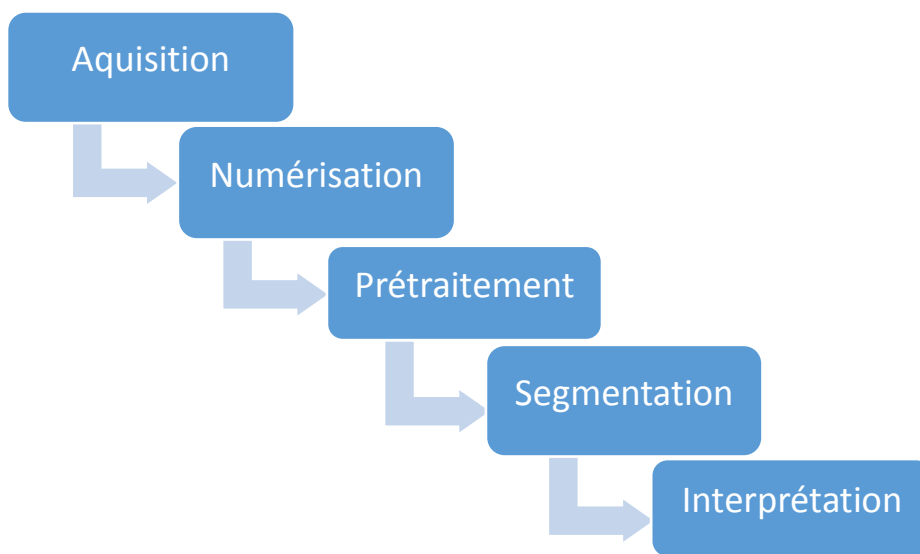


Figure I.1 : Schéma d'un système de traitement d'images

I.2. Définition de la segmentation

La segmentation de l'image est une opération de traitement d'images qui a pour but de rassembler les pixels entre eux suivant des critères prédéfinis.

Les pixels sont ainsi regroupés en régions homogènes (homogénéité de texture, de niveau de gris, de couleurs...), qui constituent une partition de l'image.

Mathématiquement, la segmentation est une décomposition de l'image I en N régions R_i . L'ensemble des régions $\{R_1, R_2, \dots, R_N\}$ est une segmentation de l'image I si **[1]**:

$$1) \bigcup_{i=1}^N R_i = I ;$$

- 2) $\forall (i, j), i \neq j, R_i \cap R_j = \phi$.
- 3) $\forall R_i, P(R_i) = \text{vrais}$.
- 4) $\forall (i, j), R_i$ spatialement adjacents à $R_j, P(R_i \cup R_j) = \text{faux}$.

La 1^{ère} condition implique que l'union de toutes les régions donne l'image originale, et que chaque pixel de l'image doit appartenir à une région. La 2^{ème} condition implique qu'un pixel ne doit pas appartenir à deux régions différentes. La 3^{ème} condition exige que chaque région doit respecter un prédicat d'uniformité notée P. La 4^{ème} condition implique la non réalisation de ce même prédicat pour la réunion de deux régions adjacentes.

Le résultat de la segmentation est une image dans laquelle une étiquette est attribuée à chaque pixel.

I.3. Approches de la segmentation d'images

La segmentation permet d'extraire d'une image une information structurale, que l'œil effectue naturellement. L'objectif est donc de découper l'image en plusieurs régions dans lesquelles les pixels vérifient un certain critère d'homogénéité.

Il existe de très nombreuses approches permettant ce découpage. Les principales sont les approches contour, les approches régions et la classification (voir figure I.2).

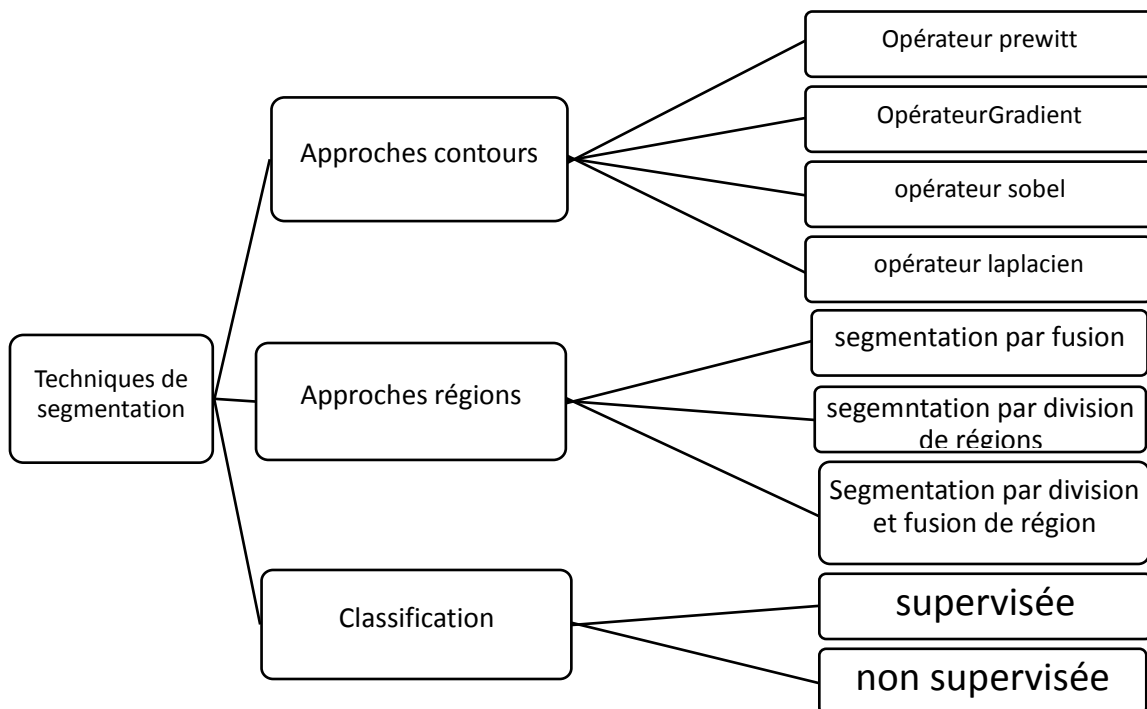


Figure I.2 : techniques de segmentation

I.3.1. Approches contour

On définit le contour comme une frontière entre deux régions de niveaux de gris différents. L'extraction des contours joue un rôle important dans tout système de vision par ordinateur. Beaucoup d'effort ont été fait pour extraire les contours d'une image et plusieurs techniques ont été développées, en utilisant principalement des opérateurs locaux de dérivation de première ordre ou du second ordre. Ces opérateurs sont suivis d'une opération de seuillage par la recherche des maxima locaux ou des passages par zéro.

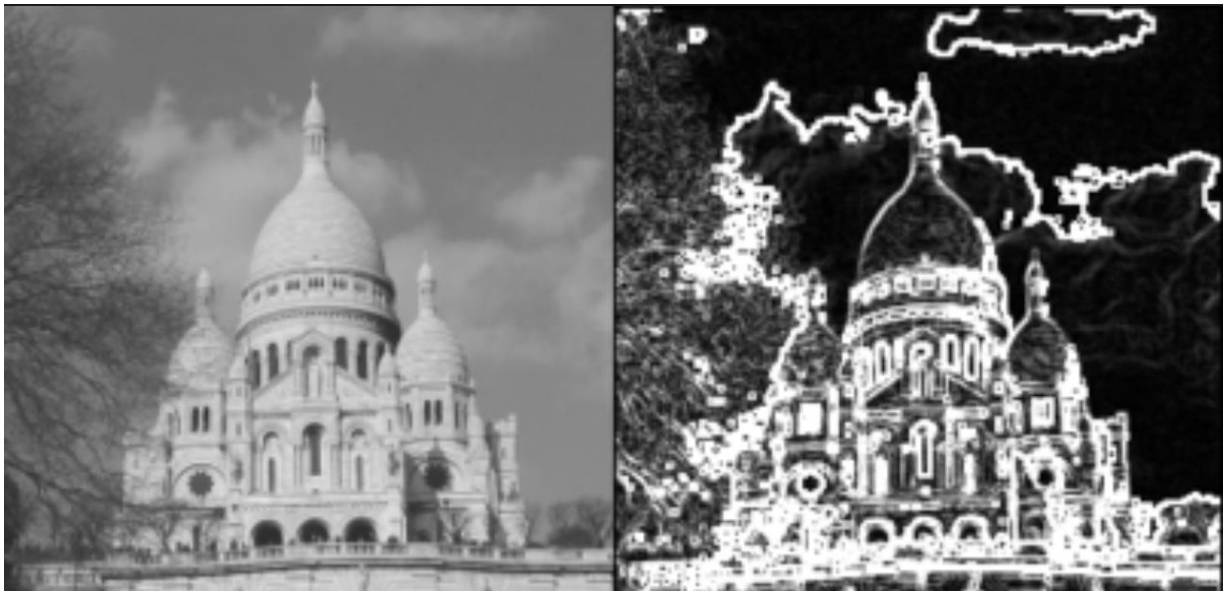


Figure I.3 : Exemple de détection de contour

Les principaux opérateurs qui sont utilisés pour l'extraction des contours sont l'opérateur gradient, l'opérateur de Laplace, l'opérateur de Prewitt et l'opérateur de Sobel [2].

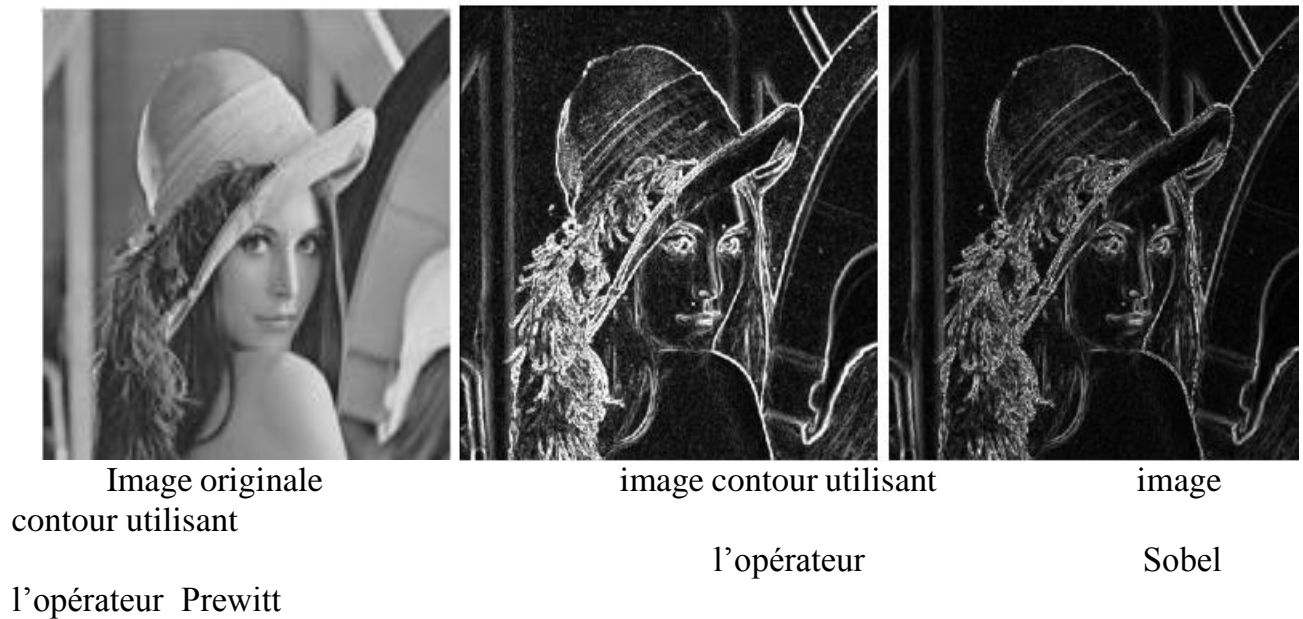


Figure I.4 : Exemple de détection de contour en utilisant les opérateurs de Sobel et Prewitt

I.3.2. Approches région

L'approche région, contrairement à l'approche contour, recherche plutôt la similarité entre les pixels. Les méthodes de cette approche fournissent une carte de régions fermée. Elle procède par un balayage de l'image afin de construire les régions.

Les trois approches région de segmentation sont:

- La segmentation par division de régions,
- La segmentation par fusion de régions,
- La segmentation par division et fusion (split and merge).

I.3.2.1. Segmentation par division de région

Ce type de méthode consiste à diviser l'image en région homogènes au sens d'un critère donné. Si le critère est valide, l'image est considérée comme segmenté.

On divise l'image en quatre blocs de taille réduits à chaque fois, le processus de divisions peut-être représenté par un arbre quaternaire.

C'est une arborescence dont la racine est l'image tout entière et tous les nœuds possèdent quatre fils. L'algorithme de découpage récursif s'arrête lorsque toutes les régions respectent le prédicat de l'homogénéité [3]. Un exemple de segmentation par division de région est donné sur la figure qui suit.

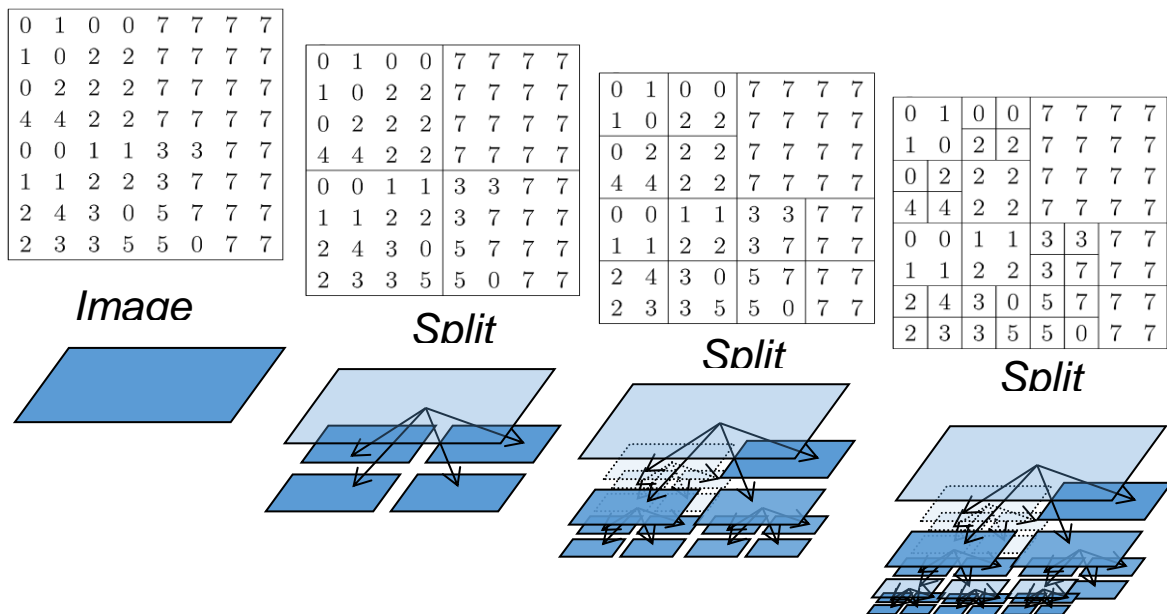


Figure I.5 : Exemple de segmentation par division de région

I.3.2.2. Segmentation par fusion

Il s'agit de faire croître des régions à partir d'un ensemble de petites régions uniformes dans l'image (de la taille d'un ou de quelques pixels), appelés germes. Puis, les régions sont construites en y ajoutant successivement les pixels qui leurs sont connexes et qui vérifient un critère d'homogénéité jusqu'à ce qu'aucun regroupement ne soit possible. Les régions construites dépendent fortement, des pixels d'amorçage choisis, de l'ordre dans lequel les pixels de la frontière sont examinés et des valeurs des seuils. La figure ci-dessous illustre un exemple de segmentation par croissance de régions.

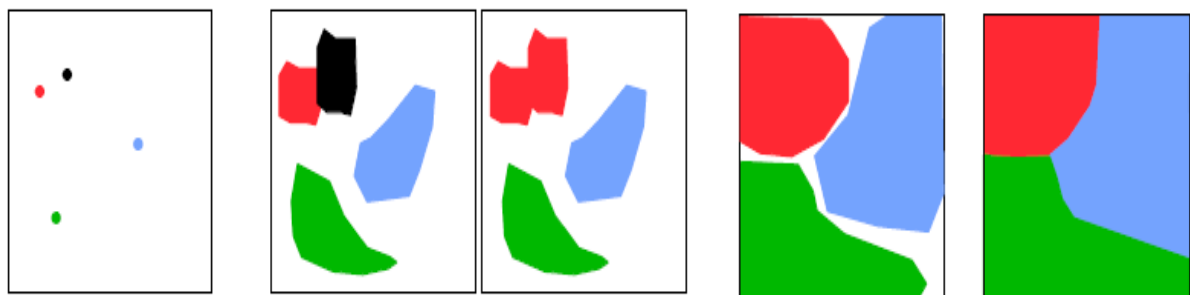


Figure I.6 : Exemple de segmentation par fusion

I.3.2.3. Segmentation par division et fusion de région (split and merge)

Son principe combine les deux méthodes précédentes. Premièrement, l'image est divisée en région homogènes. Ensuite les régions adjacentes qui répondent à des critères d'homogénéités sont fusionnées.

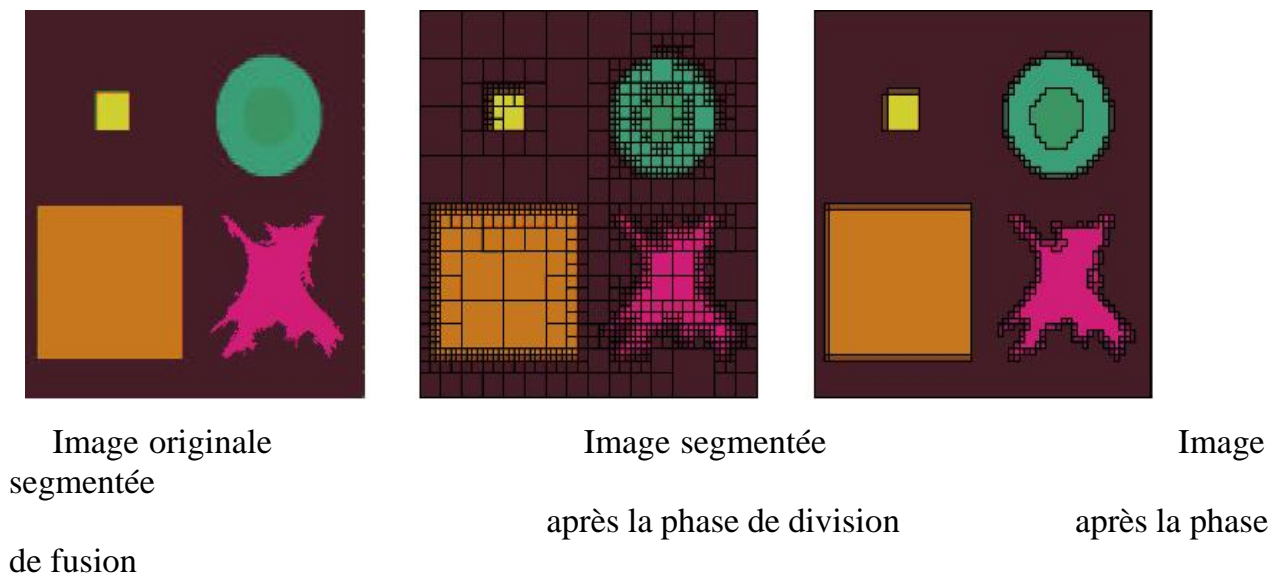


Figure I.7 : Exemple de division-fusion de région

I.3.3. Classification

La classification a toujours été un outil fort employé dans le domaine de l'analyse et du traitement d'images numériques, en particulier à des fins de segmentation ou d'interprétation.

En mathématique, On appelle classification, la catégorisation algorithmique d'objets. Elle consiste à attribuer une classe ou catégorie à chaque objet (ou individu) à classer, en se basant sur des données statistiques. Elle fait couramment appel aux méthodes d'apprentissage et est largement utilisée en reconnaissance de formes.

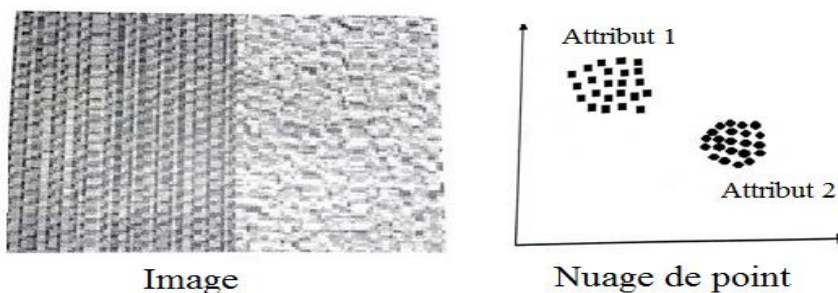


Figure I.8 : le principe de classification

Les méthodes de classification en segmentation d'images cherchent à identifier les classes des pixels présents dans l'image et affectent à chaque pixel un label (une étiquette) indiquant la classe à laquelle il appartient. Le processus de classification est donc réalisé par l'exécution des étapes suivantes :

- Représentation des données de la classification (définition des attributs des pixels servant à la classification tel que les niveaux de gris, les paramètres de textures, de couleur, ...etc).
- Définition d'une distance de similarité entre les pixels (par exemple, la distance euclidienne entre les niveaux de gris des pixels).
- Regroupement des pixels en classes.
- Evaluation de la classification obtenue.

Il existe plusieurs méthodes de segmentation par classification des pixels que l'on peut regrouper en deux catégories : classification supervisée et classification non supervisée.

I.3.3.1. Méthodes supervisées

Ce sont des méthodes dans lesquelles les classes de certains individus sont connues a priori c'est-à-dire, avant d'effectuer l'opération d'identification. Elles demandent une première phase d'apprentissage sur l'échantillon représentatif.

Cette phase d'apprentissage permet de définir des règles de décision à partir d'un ensemble d'objets de référence, dont on connaît l'identité à priori et qui sont représentatifs de chaque classe.

- Parmi ces méthodes on peut citer celles de Bayes et des K plus proches voisins (KPPV).

I.3.3.2. Méthodes non-supervisée

Contrairement à la classification supervisée cette méthode, fait référence au fait qu'aucun superviseur ou label est utilisé pour préciser à quelle classe appartient un individu. Le nombre de classes existant dans un ensemble d'individus peut être inconnu.

La classification non-supervisée (automatique) a pour but de regrouper des individus en classes homogènes, en fonction de l'analyse des caractéristiques qui décrivent les individus. Par classes homogènes, on entend collection d'objets qui sont similaires entre eux et qui sont dissemblables par rapport aux objets appartenant à d'autres groupes.

Parmi ces méthodes, on peut citer celle des k-means.

I.4. Quelques méthodes de classification :

I.4.1. Méthode k plus proches voisins (K-PPV)

La méthode des k plus proches voisins (k-PPV ou k-NN en anglais pour K-Nearest Neighbors) est supervisée et non-paramétrique. L'individu est affecté à la classe qui contient le plus d'individus parmi ces plus proches voisins. Cette méthode nécessite de choisir le nombre de voisins à prendre en compte et une distance d [11].

La distance la plus utilisée est la distance euclidienne d donnée par la relation suivante :

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^N (a_r(x_i) - a_r(x_j))^2} \quad (I.1)$$

Avec :

$d(x_i, x_j)$: distance entre le nouvel individu i et le j^{ième} individu classé,

$a_r(x_i)$: attribut du nouvel individu,

$a_r(x_j)$: attribue de j^{ième} individu classé avec $j=1, \dots, n$

et n : nombre d'individus classés,

r : indice de l'attribut,

N : nombre d'attributs.

De plus, son apprentissage est assez simple, car il est de type apprentissage par cœur (on garde tous les exemples d'apprentissage). Cependant, le temps de prédiction est très long, car il nécessite le calcul de la distance avec tous les exemples, mais il existe des heuristiques pour réduire le nombre d'exemples à prendre en compte.

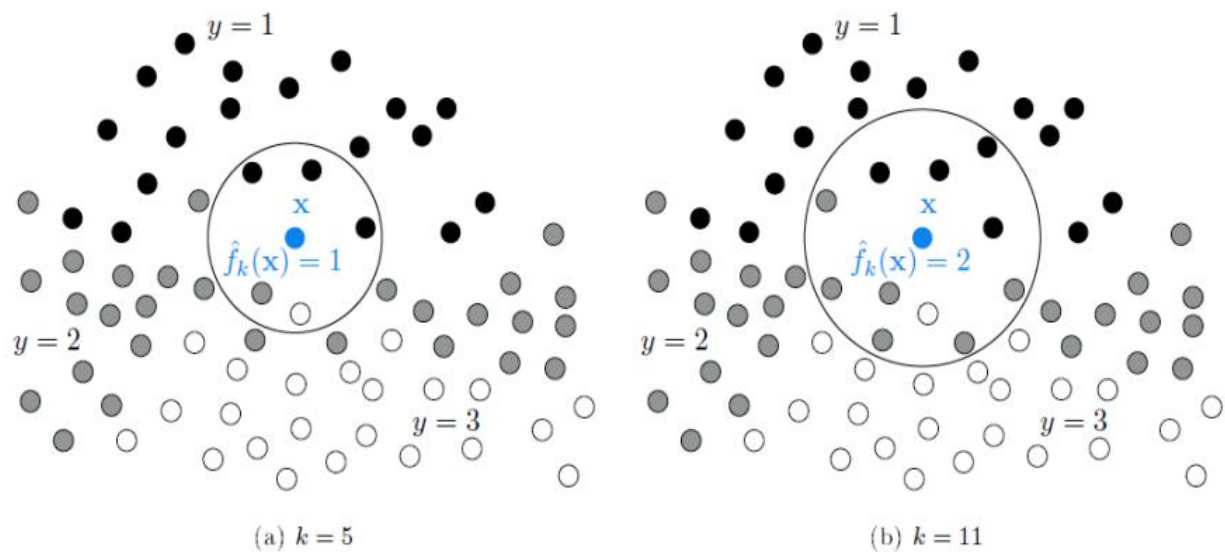


Figure I.9 : Exemple de classification avec les **K-PPV** (K=5 et k=11)

Dans la figure précédente, à gauche, la classification est simple quel que soit le nombre de voisins choisis : le nouvel objet est noir. A droite, en revanche, tout dépend du nombre de voisins choisis et de l'heuristique de classification [4].

Pour $k = 1$, le nouvel objet est gris. Pour $k= 3$, si les trois voisins ont le même poids, alors le nouvel objet est noir. Par contre, si le poids est pondéré par l'inverse de la distance alors le nouvel objet peut être gris. Cela revient à pondérer l'affectation de classe avec la distance : plus un voisin est éloigné, plus son influence est faible.

Les principaux inconvénients de cette méthode sont le nombre d'opérations nécessaires pour classer une entité dans le cas d'une grande base de référence ainsi que sa sensibilité au bruit présent dans les données d'apprentissage.

I.4.2. Méthodes k-médoids

Dans des méthodes de k-médoids une classe est représentée par un de ces individus (médoids). C'est une méthode itérative combinant la réaffectation des individus dans des classes avec une intervention des médoids et des autres individus [6].

C'est une méthode simple parce qu'elle couvre n'importe quel type de variables, quand des médoids sont choisis des classes sont définis comme sous-ensembles des individus près des médoids les plus proches par rapport à une mesure de distance choisie.

Donc il est plus judicieux de choisir comme centre de groupe un individu présent dans le groupe et non un individu calculé, le médoids de groupe est l'individu possédant la dissimilarité moyenne la plus faible par rapport aux autres individus du groupe.

Ces méthodes ont une bonne résistance aux données erronées et sont flexibles avec tout type de distance.

Elles nécessitent de spécifier le nombre de classes k .

I.4.3. Méthode des k-means

L'algorithme des K-means est l'une des techniques de classification non supervisée les plus utilisées. La méthode consiste à placer aléatoirement dans l'espace K "centroids" afin de déterminer K classes. On affecte à chacun de ces "centroids" les données les plus proches, puis on calcule la position moyenne des données associées aux "centroids" que l'on déplace en ce point. Les opérations d'affectation d'objet et de déplacement du centroid sur la moyenne sont répétées jusqu'à ce que chaque centroid ait atteint une position stable.

Malgré sa simplicité, cet algorithme se révèle efficace. Toutefois il est nécessaire de prédéterminer le nombre de catégories et la position de départ des prototypes qui a un impact sur le découpage en classes.

L'algorithme K-means est donné ci-dessous [5] :

Entrée

Ensemble de N données, noté par x

Nombre de classes souhaité, noté par k

Sortie

Une partition de K groupes $\{C_1, C_2, \dots, C_k\}$

Début

1) Initialisation aléatoire des centres C_k ;

Répéter

2) Affectation : générer une nouvelle partition en assignant chaque pixels au groupe dont le centre est le plus proche ;

$$x_i \in C_k \text{ si } |x_i - \mu_k| \leq \min_j |x_i - \mu_j|$$

Avec μ_k le centre de la classe k .

3) Représentation : Calculer les centres associe à la nouvelle partition .

$$\mu_k = \frac{1}{N} \sum_{x_i \in C_k} x_i$$

Jusqu'à convergence de l'algorithme vers une partition stable.

FIN

Les principales étapes de cette algorithmme sont données par la figure (I.12) est sont les suivantes [11] :

Itération 0 : on tir k individus au hasard, ces individus représentent k centres provisoires de classes :

$$c^0 = \{c_1^0, c_2^0, \dots, c_k^0\}$$

Itération 1 : on calcule les distances qui séparent chaque individu des centres des classes. Chaque individu est affecté à la classe du centre le plus proche. On obtient ainsi une première partition P^0 de l'ensemble en k classes :

$$p^0 = \{p_1^0, p_2^0, \dots, p_k^0\}$$

Itération 2 : on calcule donc pour chaque classe de P obtenue précédemment les k nouveaux centres de classes :

$$C^1 = \{c_1^1, c_2^1, \dots, c_k^1\}$$

Puis on réaffecte les individus aux centres les plus proches. Ce qui permet de définir une nouvelle partition P^1 de l'ensemble.

Itération m : à cette étape, on détermine les k centres $c_1^{m-1}, c_2^{m-1}, \dots, c_k^{m-1}$ des classes respectives de la partition de l'ensemble obtenu à l'étape (m-2) soit :

$$p^{(m-1)} = \{p_1^{(m-1)}, p_2^{(m-1)}, \dots, p_k^{(m-1)}\}$$

Le processus est réitéré jusqu'à ce que deux itérations successive conduisent à la même partition, ou le nombres d'itérations choisi par l'utilisateur est atteint.

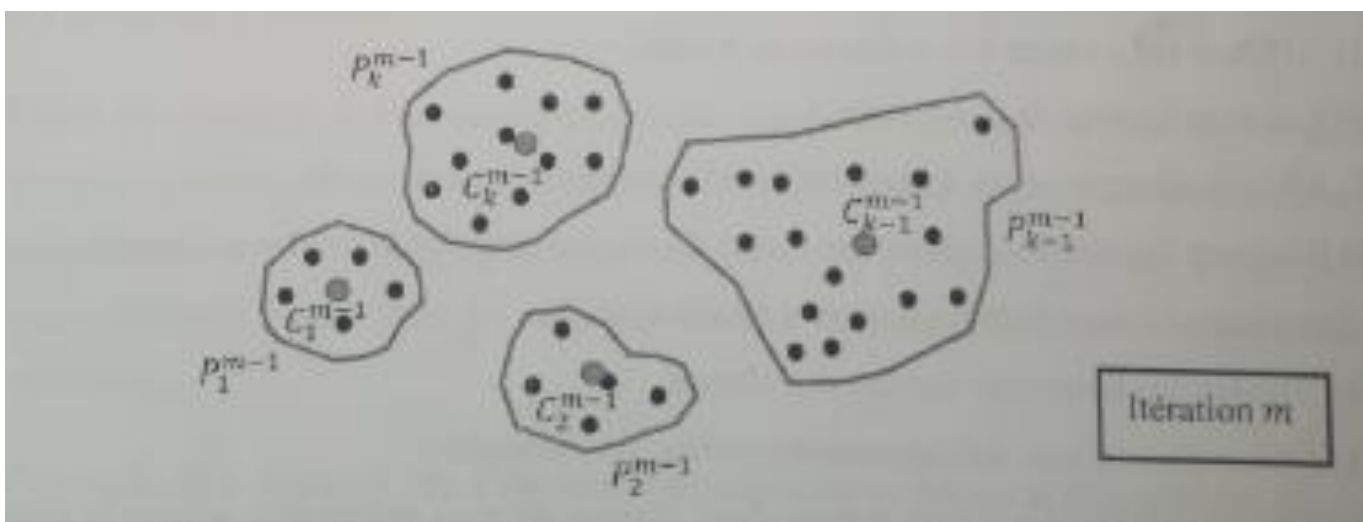
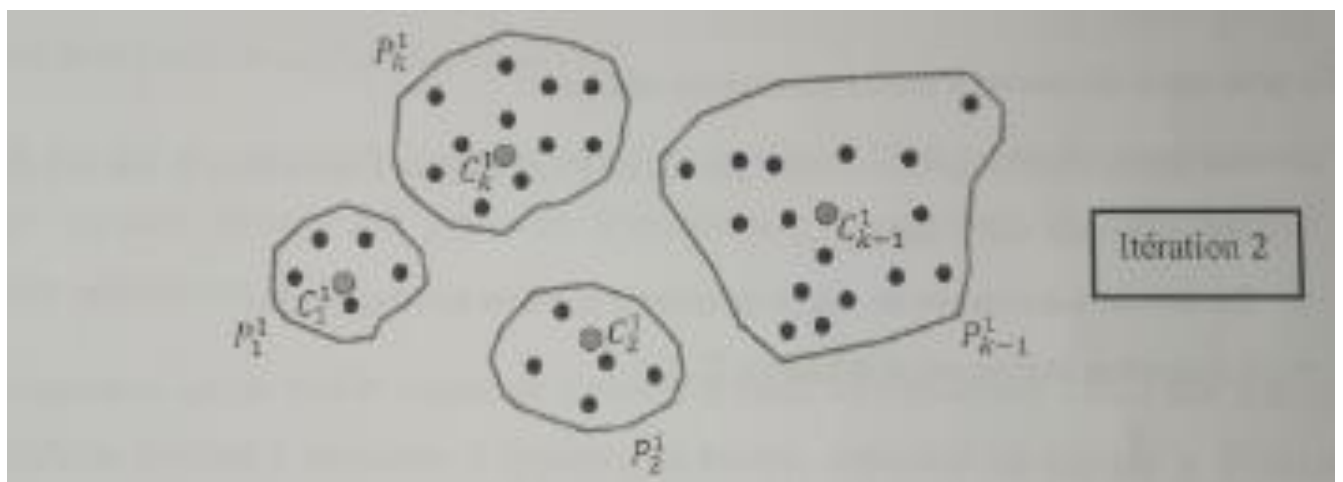
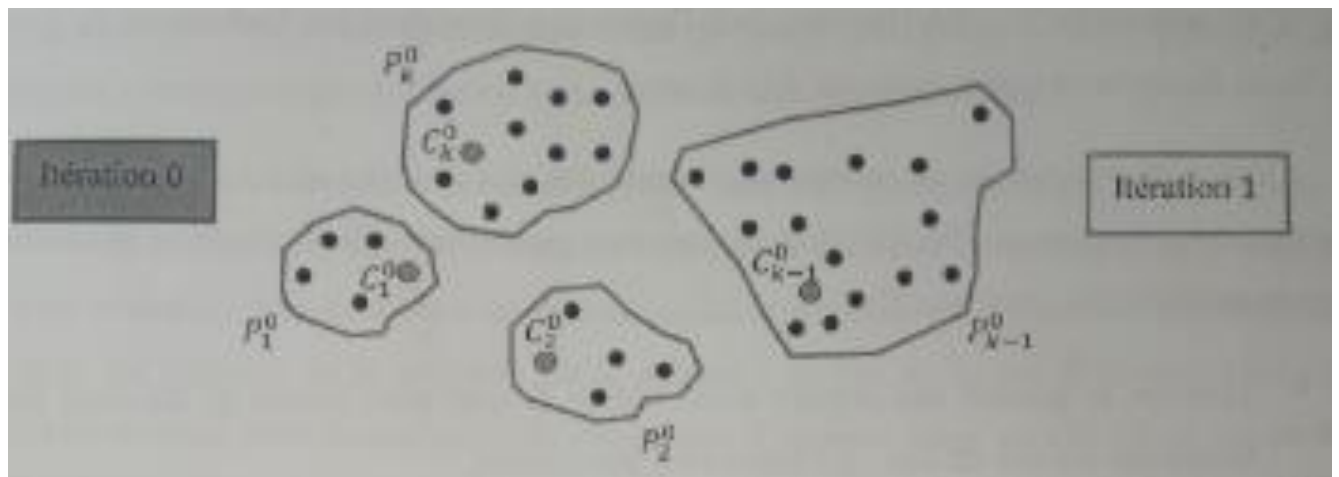


Figure I.10. Déroulement de l'algorithme de k-means

I.5. Conclusion

Dans ce chapitre nous avons présentée d'une manière générale la segmentation d'image et ses différentes approches. Nous avons plus particulièrement détaillé la méthode des k-means qui présente l'avantage d'être simple et rapide.

Cette méthode sera utilisée dans les prochains chapitres, afin de segmenter des images simples et des images superpixelisées.

CHAPITRE II: **La superpexilisation**

II.1. Introduction

De nombreuses applications de traitement d'images peuvent bénéficier d'une étape de pré-segmentation groupant a priori les pixels de l'image. Par exemple, l'analyse d'images consiste généralement à reconnaître et localiser les différents objets présents dans une image. Cette reconnaissance peut se faire par une segmentation préalable des objets, qui sont ensuite classés. La segmentation est donc une étape essentielle qui peut avoir un impact fort sur la précision ainsi que le temps de calcul de la méthode de classification.

Afin de réduire le temps de calcul, des méthodes de décomposition en superpixels ont été introduites. Ces méthodes regroupent les pixels de l'image en zones homogènes tout en essayant de respecter les contours des objets de l'image, réduisant ainsi le nombre d'éléments à traiter, et par conséquent la charge de calcul.

II.2. Définition de la superpexilisation

Une superpexilisation, est une sur-segmentation basée généralement sur les couleurs ou bien les niveaux de gris des pixels de l'image.

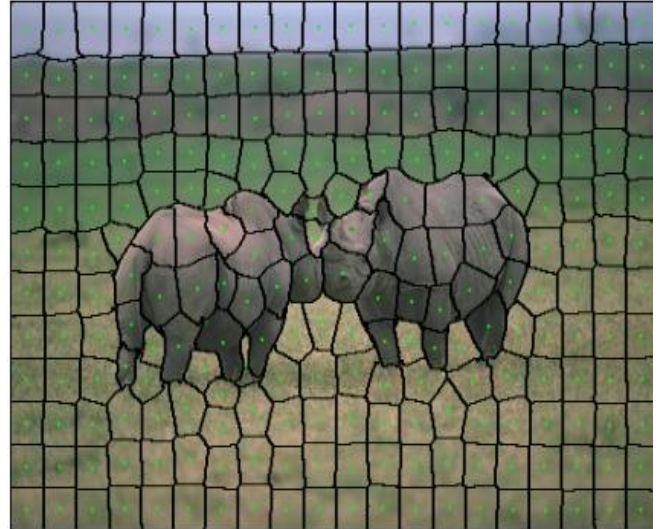
Un superpixel est alors défini comme un groupe de pixels ayant des caractéristiques similaires.

La superpexilisation est donc une simplification de l'image. Elle réduit le nombre de données à traiter réduisant ainsi les coûts de calculs des différentes opérations et améliore les performances par rapport à la méthode basée pixels.

Les méthodes des superpexilisation peuvent se diviser en deux catégories qui produisent des décompositions soit régulières ou bien irrégulières (voir figure II.1) [10].



(a) Décomposition irrégulière



(b) Décomposition régulière

Figure II.1 : Exemple d'image régulière et irrégulière

Les méthodes irrégulières produisent des superpixels qui peuvent être de tailles très différentes et avoir des frontières bruitées (voir la figure II.1.a).

Les méthodes régulières produisent des superpixels de tailles équivalentes. Elles permettent de calculer des descripteurs plus robustes (voir figure II.1.b).

Pour la plupart de ces méthodes un compromis est calculé entre la régularité de la décomposition, l'homogénéité couleur ou niveau de gris des superpixels et l'adhérence aux contours de l'image.

Pour une superpexilisation irrégulière, les intensités des pixels contenus dans chaque superpixel ont une très faible variance, contrairement aux superpixels d'une décomposition régulière qui ne s'adapte pas au contenu de l'image. Cette comparaison entre décomposition régulière et irrégulière avec les superpixels est illustrée dans la figure suivante.

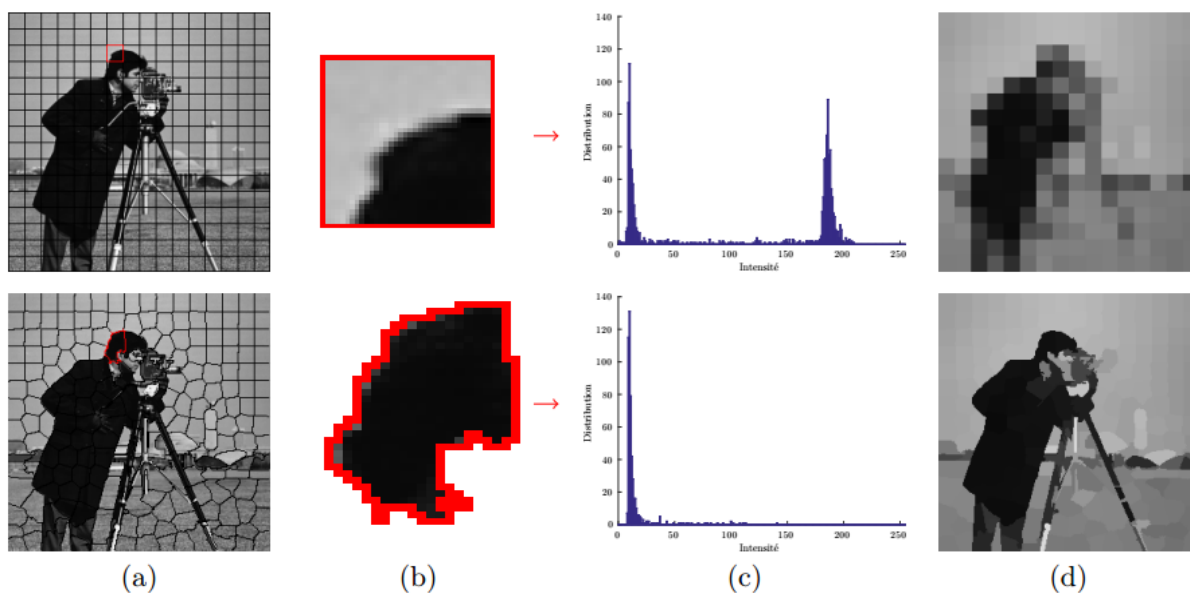


Figure II.2 : Illustration de la faible variance d'intensité dans les superpixels

L'image *Cameraman* est décomposée en blocs réguliers (en haut) et en superpixels avec la méthode SCALP (en bas). Un superpixel est sélectionné pour chaque décomposition en (b) et l'histogramme des intensités des pixels contenus dans ce superpixel est donné en (c). L'intensité moyenne dans chaque superpixel est alors représentée en (d). Les superpixels permettent de résumer efficacement le contenu de l'image.

II.3. Méthodes de décomposition en superpixels

De très nombreuses méthodes de décomposition en superpixels ont été proposées au fil des années. Les plus populaires sont : les méthodes de graphe, hiérarchiques et de montée de gradient [9].

II.3.1. Méthodes de graphes

La représentation sous forme de graphe est utilisée dans différentes applications d'analyse d'images. Dans ces applications, l'image est traitée pour produire un graphe représentant les composants et les relations entre eux. Les pixels de l'image sont les nœuds du graphe et les liens reliant chaque pixel à ses voisins sont les arcs (voir la figure II.3). En résumé, pour construire le graphe les nœuds et les arcs peuvent avoir des étiquettes (labels). La valeur associée à chaque nœud représente son niveau de gris, tandis que celle associée à chaque arête correspond à la différence de niveaux de gris des nœuds qu'elle relie [12].

La plupart des méthodes de segmentation d'images à base de graphes favorisent le regroupement de pixels adjacents et similaires suivant certains critères de continuité dans l'image (similarité, proximité...etc.).

Ces approches permettent aussi bien de produire des superpixels réguliers ou irréguliers.

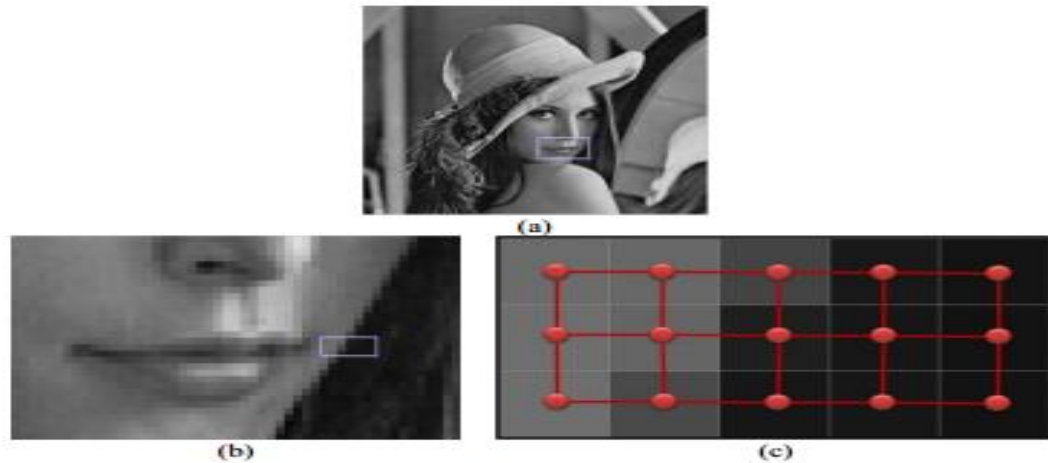


Figure II.3: Graphe d'adjacence de pixels : (a) Image originale (*Lena*), (b) Zoom sur la région sélectionnée dans (a), (c) Chaque pixel de l'image correspond à un nœud du graphe, les arcs sont construits via un système d'adjacence de voisinage 4

II.3.2. Méthodes hiérarchiques

Les approches de type hiérarchique comme SEEDS (Superpixel Extracted via Energy-Driving Sampling. [09]), ETPS (Extended Topology Preserving Superpixels. [10]) considèrent l'image à différentes échelles et convergent progressivement vers une décomposition à l'échelle pixellique. De façon similaire à une approche de type arbre quaternaire, elles segmentent itérativement l'image en considérant des blocs de taille de plus en plus réduite jusqu'à ne contenir qu'un pixel. Ces approches sont initialisées depuis une décomposition régulière en blocs larges et raffine les bordures des superpixels en effectuant des transferts de blocs.

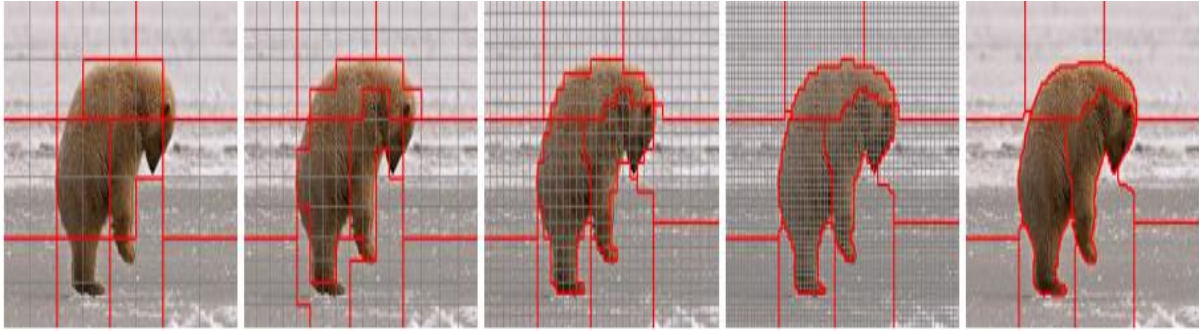


Figure II.4 : Illustration du processus de décomposition de la méthode ETPS de type hiérarchique

En général les approches blocs de types hiérarchique peuvent facilement être optimisées numériquement et permettent d'atteindre de faibles temps de calcul en concentrant le raffinement de la décomposition sur les blocs en bordure des superpixels.

II.3.3. Méthodes de montée de gradient

Elles s'inspirent des méthodes connues des k-means, et du mean-shift. Initialement utilisées dans le contexte de la classification, elles ont été introduites dans des approches de construction de superpixels telles que quick shift ou SLIC.

Dans l'ensemble, les méthodes de montée de gradient permettent à la fois de produire des superpixels très réguliers et très irréguliers. Pour la majorité de Ces méthodes présentent l'inconvénient majeur de ne pas permettre le contrôle du nombre de superpixels générés.

II.3.4. Comparaison des méthodes de décomposition en superpixels

Le tableau II.1 regroupe les méthodes de décomposition en superpixels les plus connus et informe sur le type de la technique, sa capacité à régler le nombre de superpixels en sortie et à fournir des superpixels réguliers

	FH	QS	TP	ERS	SEEDS	SLIC	ERGC	WP	LSC	ETPS	SCALP
Type	bg	mg	mg	bg	h	mg	bg	mg	mg	h	mg
Nombre	✗	✗	✓	✓	✓	✗	✓	✓	✓	✓	✗
Régularité	✗	✗	✗	✗	✗	✓	✓	✓	✓	✓	✓

Tableau II.1 : Comparaison des méthodes de superpixels de l'état-de-l'art basées sur un graphe (bg), hiérarchiques (h) ou de montée de gradient (mg). Une vérification (□) nous indique la capacité des méthodes à régler le nombre de superpixels en sortie et leur régularité [10].

Les techniques citées dans le tableau sont [09] [10] [11]:

FH : La méthode de Felzenszwalb et Huttlocher

QS : la méthode de Quick Shift

SCALP: Superpixels with Contour Adherence using Linear Path.

SLIC: Simple Linear Iterative Clustering.

ETPS: Extended Topology Preserving Superpixels

TP : méthode de Tanimoto et Pavlidis.

WP: méthode de Water pixels.

LSC: Linear spectral clustering superpixel.

SEEDS: Superpixel Extracted via Energy-Driving Sampling.

ERGC: Eikonal-based Region Growing Clustering

ERS : Entropy Rate Superpixel

Nous allons développer plus bas deux de ces techniques.

II.4. Méthode SLIC : (simple linéaire itératif clustering).

L'algorithme de SLIC est sans doute l'une des méthodes les plus célèbres pour découper une image en superpixels. C'est une méthode qui, par défaut, n'exige que le nombre K des superpixels à générer. Elle regroupe les pixels de l'image en des superpixels ayant des tailles presque identiques.



Figure II.5 : Exemple de superpexilisation par l'algorithme SLIC, les contours des superpixels sont représentés en rouge

L'algorithme de SLIC est détaillé ci-dessous [11].

1- Initialiser les centres des superpixels

$$C_k = [C1_k, C2_k, C3_k, x_k, y_k]$$

Avec :

- $k = 1, 2, 3, \dots, K$ ou K est le nombre des superpixels
- $C1_k, C2_k, C3_k$: sont les niveaux des composantes couleurs moyennes de tous les pixels qui se trouvent dans le carré de taille SXS
- x_k, y_k sont choisies dans le carré de tailles SXS en prenant le centre d'une fenêtre de taille $3X3$ ou le gardien est le plus faible.

2- Créer une carte des étiquettes l et l'initialiser en posant :

$$l(i) = -1 \text{ pour tous les pixels } i \text{ de l'image.}$$

3- Créer une carte des distances d et l'initialiser en posant :

$$d(i) = \infty \text{ pour tous les pixels } i \text{ de l'image.}$$

4- pour chaque centre de superpixels C_k , faire :

- Pour chaque pixel i se trouvant dans la région $(2S+1)X(2S+1)$ autour de C_k ,

faire :

- Calculer la distance D entre C_k et i .

- si $D < d(i)$ alors $d(i) = D$ et $l(i) = k$.

Fin pour

Fin pour

5- Mettre à jour les positions et les couleurs moyennes des centres des superpixels.

6- Refaire les étapes 4 et 5 jusqu'à ce que le déplacement moyennes des centres entre deux itérations soit inférieur à un seuil.

7- Renforcement de la connexité .

Donc la méthode SLIC commence par la décomposition de l'image en une grille régulier composée de blocs de taille SXS avec :

$$S = \sqrt{\frac{N}{K}} \quad (\text{II.1})$$

où N est le nombre de pixels dans l'image et K est le nombre de superpixels à générer.

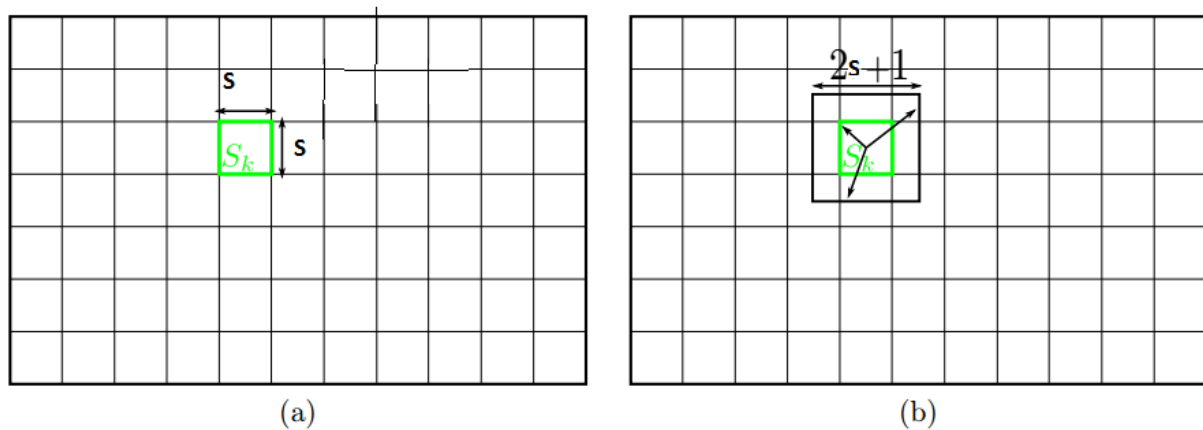


Figure II.6 : Illustration de la méthode de groupement itératif SLIC (Achanta et al., 2012). (a) Initialisation des superpixels S_k avec une grille régulière en blocs de taille $S \times S$. (b) Zone de recherche d'association des pixels de taille $(2S + 1) \times (2S + 1)$ centrée sur le barycentre spatial du superpixel

Ensuite, pour chaque superpixel, on calcule son centre c_k qui est à cinq dimensions (voir l'algorithme). Trois composantes sont les composantes couleurs qui sont souvent les composantes de l'espace de représentation Lab ($C1_k = L_k, C2_k = a_k, C3_k = b_k$). Les deux autres composantes sont les composantes spatiales qui donnent la position des centres dans l'image.

Puis pour chaque centre, on lui réaffecte des pixels en limitant la taille de la zone de recherche à $(2S+1) \times (2S+1)$ autour de ce centre de superpixel.

Cette limitation de la taille de la zone de recherche a pour avantage de réduire le nombre de calculs de distances, ce qui rend l'algorithme rapide.

Lorsque les pixels ont été associés à leur centres de superpixels, les centres des superpixels sont mis à jour.

Les phases d'affectation et de mise à jour sont répétées de façon itératives jusqu'à ce que les centres ne se déplacent quasiment pas.

La dernière étape consiste à renforcer la connexité en fusionnant des superpixels de taille non significative (seuil à déterminer) à des superpixels voisins est de taille significative.

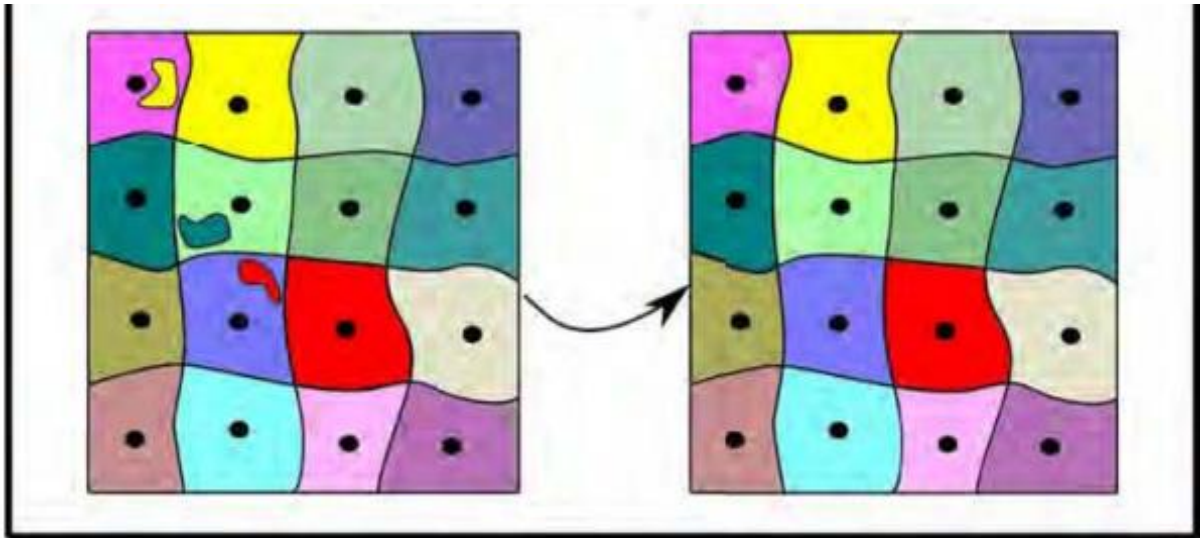


Figure II.7 : Exemple de renforcement de la connexité

Les affectations des pixels aux centre des superpixels sont réalisées après calcul d'une distance D.

D mesure donc la distance entre un pixel $i = (C1_i, C2_i, C3_i, x_i, y_i)$, et un centre de superpixel $C_k = [C1_k, C2_k, C3_k, x_k, y_k]$.

La couleur d'un pixel est représentée par trois composantes (C1, C2, C3) dont la gamme des valeurs possibles est connue. La position du pixel (x, y) peut prendre un ensemble de valeurs qui varie selon la taille de l'image.

Si D était simplement défini comme la distance euclidienne à 5 dimensions cela causerait des problèmes de classification pour différentes tailles de superpixels. Pour les grands superpixels par exemple, les distances spatiales l'emporteraient sur les proximités de couleur [7].

Afin de combiner les deux distances en une unique mesure, il est nécessaire de normaliser la proximité de couleur et la proximité spatiale par leurs distances maximales respectives N_c et N_s dans un superpixel.

La proximité de couleur est donnée par :

$$dc = dc(i, C_k) = \sqrt{(C1_k - C1_i)^2 + (C2_k - C2_i)^2 + (C3_k - C3_i)^2} \quad \text{(II.2)}$$

La proximité spatiale est donnée par:

$$ds = ds(i, C_k) = \sqrt{(x_k - x_i)^2 + (y_k - y_i)^2} \quad \text{(II.3)}$$

la distance D est donnée par:

$$D = dc + \frac{m^2}{s^2} ds \quad (\text{II.4})$$

où D est la somme de la distance de l'espace couleur et de la distance spatiale normalisée par l'intervalle de grille S. Une variable m est introduite dans D permettant de contrôler la compacité du superpixel. Plus la valeur de m est grande, plus la proximité spatiale est importante et plus le superpixel est compact.

A l'inverse, si m est petit les superpixels adhéreront plus aux contours mais auront une forme moins régulière. D'autres mesures de distance D existent, parmi elle nous pouvons citer trois qui sont :

$$D = \sqrt{\left(\frac{dc}{Nc}\right)^2 + \left(\frac{ds}{Ns}\right)^2} \quad (\text{II.5})$$

$$D = \sqrt{\left(\frac{dc}{m}\right)^2 + \left(\frac{ds}{s}\right)^2} \quad (\text{II.6})$$

$$D = \sqrt{dc^2 + \left(\frac{ds}{s}\right)^2 * m^2} \quad (\text{II.7})$$

La distance spatiale maximale à l'intérieur d'un superpixel correspond à l'intervalle d'échantillonnage $Ns = S = \sqrt{\frac{N}{k}}$. La distance de couleur maximale Nc est plus difficile à estimer car elle peut varier d'un superpixel à un autre et d'une image à une autre. Le problème est de l'éviter, en fixant Nc à une valeur constante.

II.5 La méthode SCALP

Dans cette partie, nous développons une méthode de décomposition en superpixels appelée SCALP (Superpixels with Contour Adhérence using Linear Path). Cette méthode des superpixels est basée sur l'algorithme SLIC [9], sauf que la distance calculée dans le but d'associer un pixel à un superpixel pendant le regroupement est modifiée.

Un chemin linéaire du pixel en question au centre du superpixel est introduit. Ce chemin permet le calcul d'une distance couleur plus précise qui améliore l'homogénéité des couleurs et la régularité des superpixels.

Une information de contour dans ce chemin est aussi prise en considération pour assurer le respect des objets de l'image.

La méthode SCALP est résumée dans la figure II.8

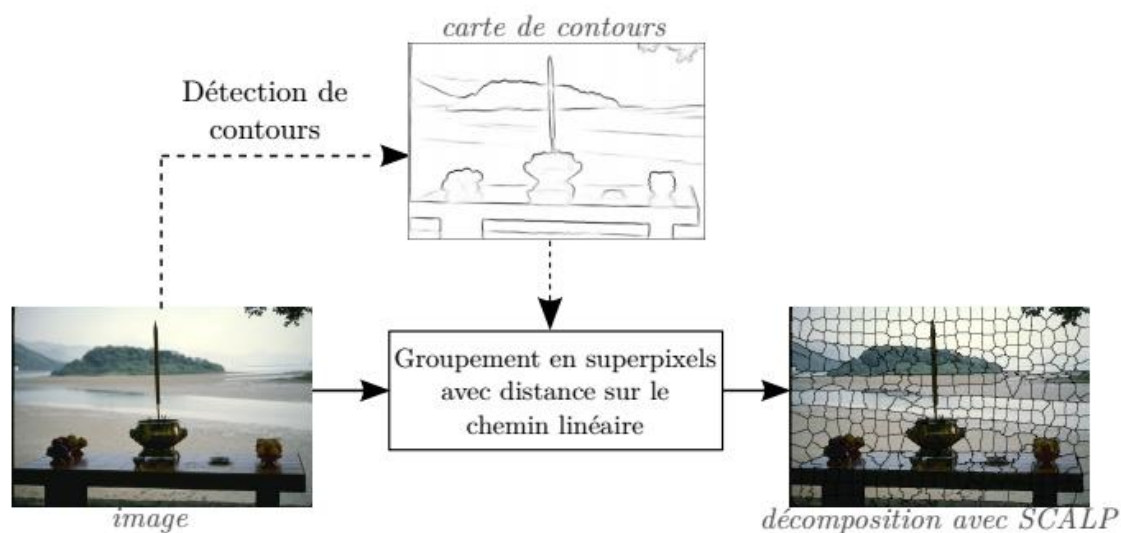


Figure II.8 : La méthode SCALP. Une carte de contours peut être utilisée pour améliorer le respect des contours de l’image

Avant de donner l’algorithme de la méthode SCALP on définit le chemin linéaire puis la nouvelle distance et l’influence de la carte contour.

II.5.1 Chemin linéaire

On définit le chemin p_i^k contenant les pixels entre la position $X_i = (x_i, y_i)$ d’un pixel i et le centre $X_k = (x_k, y_k)$ d’un superpixel S_k . Afin de limiter le nombre de pixels dans le chemin p_i^k , on ne considère que les pixels sur le chemin linéaire entre X_i et X_k , comme cela est illustré dans la figure II.9.

Les pixels considérés q (rouge) sont ceux qui intèrèsectent avec le segment (flèche)

entre X_i , et X_k . Considérer ce chemin linéaire permet de limiter le temps de calcul et d'augmenter naturellement la régularité de la décomposition [8].

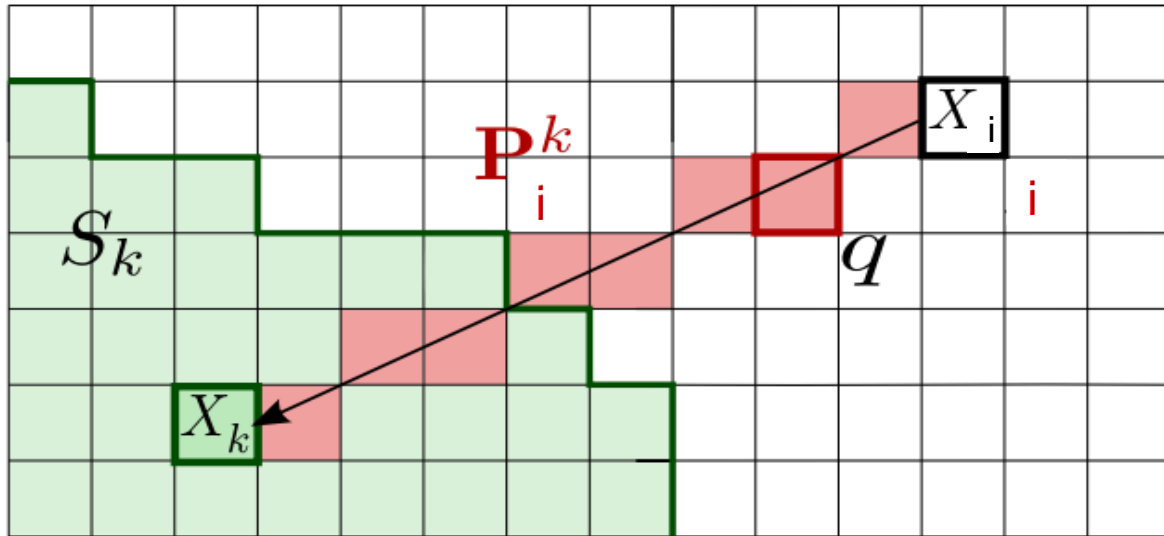


Figure II.9 : Illustration du chemin linéaire P_i^k entre un pixel i et un superpixel S_k de centre X_k .

II.5.2 Distance couleur au superpixel

La méthode SCALP peut renforcer la précision de la distance couleur en la calculant également sur P_i^k , l'ensemble des pixels q sur le chemin entre la position du pixel i et le centre du superpixel S_k , donc la nouvelle distance de couleur est définie comme suit :

$$d_c(i, C_k, P_i^k) = \lambda d_c(i, C_k) + (1 - \lambda) \frac{1}{|P_i^k|} \sum_{q \in P_i^k} d_c(q, C_k) \quad (\text{II.8})$$

où $\lambda \in [0, 1]$ pondère l'influence de la distance couleur le long du chemin entre X_i et X_k . SCALP améliore l'homogénéité couleur, puisque pour que la distance II.8 soit faible, la couleur sur P_i^k doit être proche de la couleur moyenne du superpixel. La distance de l'équation II.8 empêche également l'apparition de formes irrégulières et améliore naturellement la régularité [8].

II.5.3 Adhérence à la carte contour

L'homogénéité de couleur est la régularité des superpixels améliorées par la distance $d_c(i, C_k, P_i^k)$ peut ne pas correspondre au respect des objets de l'image. Une carte de

contour peut être exploitée sur le chemin linéaire afin d'augmenter l'adhérence des superpixels aux contours. Cette carte notée δ , est telle que :

$$\delta(i) = \begin{cases} 1 & \text{si un contour est détecté au pixel } i \\ 0 & \text{sinon} \end{cases}$$

La distance D est pondéré par la valeur $dc(p_i^k)$ donnée par la relation :

$$dc(p_i^k) = 1 + \gamma \max_{q \in P_i^k} \delta(q) \quad (\text{II.9})$$

où $\gamma \geq 0$ pondère l'influence de la carte de contours sur P_i^k .

La Figure II.8 illustre la sélection du maximum d'amplitude de contour sur le chemin linéaire. Lorsqu'une forte amplitude de contour se trouve sur le chemin entre i et le barycentre de S_k , l'équation II.8 empêche ce pixel d'être associé au superpixel.

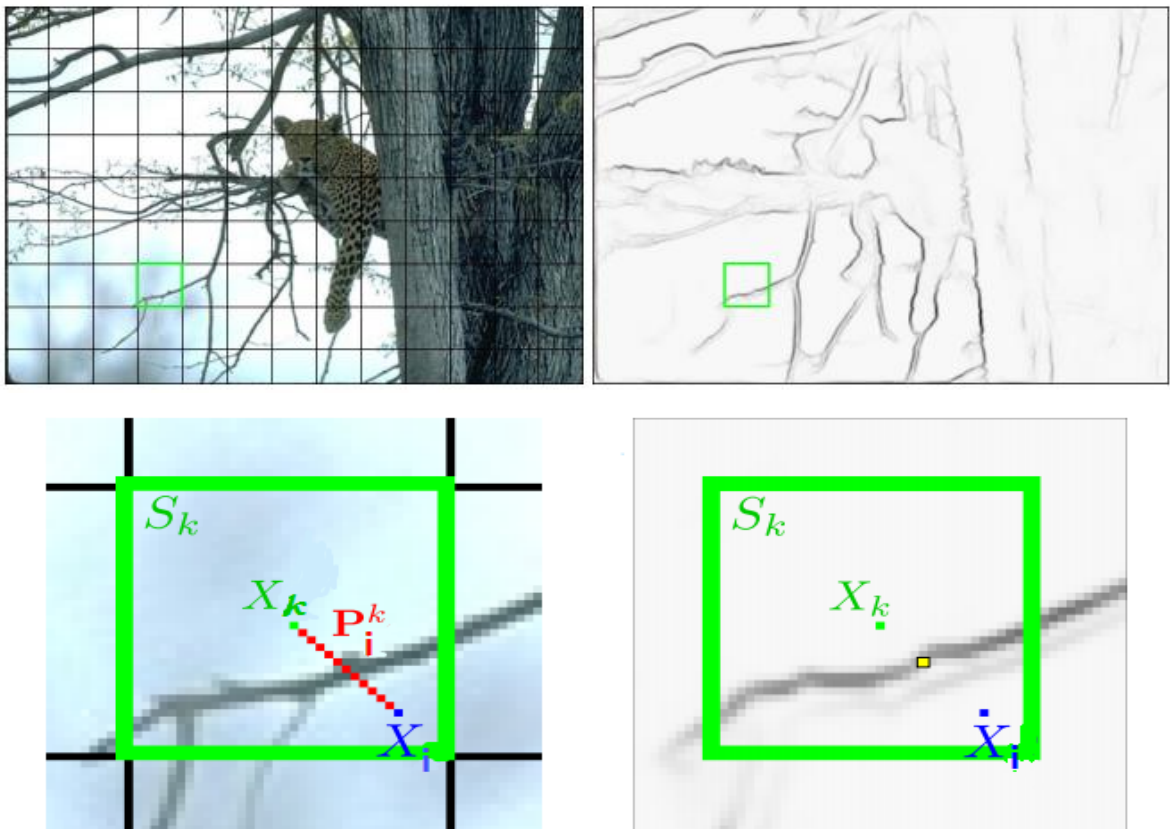


Figure II.10 : Illustration de la première itération de la méthode SCALP

- a) Grille régulière initiale,
- b) Carte de contours,
- c) Chemin linéaire P_i^k ,
- d) Contour maximal sur P_i^k .

Pour assurer le respect des contours, ce terme $dc(\mathbf{P}_i^k)$ est multiplié aux distances couleur et spatiale pour définir la distance finale D à minimiser durant le processus itératif de groupement :

$$D(i, C_k) = (dc(i, C_k, \mathbf{P}_i^k) + ds(i, C_k) \frac{m^2}{s^2}) dc(\mathbf{P}_i^k), \quad (II.10)$$

avec la distance spatiale ds est calculée comme dans SLIC (voir l'équation II.3), et m est le paramètre de la régularité.

Un m élevé force la convexité et produit des superpixels carrés pour une grille initiale régulière, quand un m faible permet une meilleure adhérence aux contours, mais produit des superpixels aux formes irrégulières.

II.5.4 algorithme SCALP

La méthode SCALP est résumée dans l'Algorithme suivant pour une image I décomposée en K superpixels.

- 1-Initialisation des centres des superpixels depuis une grille régulière.
- 2-Créer une carte des étiquettes l et l'initialiser en posant :
 - $l(i) = 0$ pour tous les pixels i de l'image.
- 3- Créer une carte des distances d et l'initialiser en posant :
 - $d(i) = \infty$ pour tous les pixels i de l'image.
- 4- Calculer la carte de contour δ (équation II.9)
- 5- Pour chaque centre de superpixel c_k , faire :
 - Pour chaque pixel i se trouvant dans la région $(2S+1) \times (2S+1)$ autour de c_k , faire :
 - Calculer le chemin linéaire P_i^k .
 - Calculer la distance D entre c_k et i en utilisant l'équation II.10
 - Si $D < d(i)$ alors $d(i) = D$ et $l(i) = k$.
 - Fin pour chaque pixel i
 - Fin pour chaque centre c_k
- 6- Mettre à jour les positions et les couleurs moyennes des centres des superpixels.
- 7- Refaire les étapes 5 et 6 jusqu'à ce que le déplacement moyen des centres entre deux itérations soit inférieur à un seuil.
- 8- - Renforcement de la connexité.

II.6 Conclusion

Dans ce chapitre nous avons présenté un aperçu sur deux méthodes de décomposition en superpixels, qui permettent de diviser l'image en zones homogènes réduisant ainsi le nombre d'éléments à traiter afin de diminuer le temps de calcul.

Dans le chapitre suivant nous allons appliquer une de ces méthodes sur des images avant de les segmenter avec l'algorithme k-means. Une analyse des résultats obtenus sera effectuée.

CHAPITRE III: **Tests et résultats**

III.1. Introduction

Pour réaliser la segmentation d'images plusieurs algorithmes sont proposés dans la littérature. Parmi les méthodes de segmentation par classification, celle utilisant l'algorithme k-means est très utilisée.

La segmentation des images en utilisant l'algorithme k-means consiste à classer tous les pixels de l'image en un nombre de classe imposée par l'utilisateur.

Cette technique peut prendre un temps de calcul important surtout pour des images de grandes dimensions. Nous proposons de réduire ce temps de calcul en réalisant une superpexilisation au préalable.

La technique de superpexilisation choisie est la technique SLIC « Simple Linéaire Itératif Clustering ». Dans cette dernière, le nombre de superpixels est choisi par l'utilisateur.

Nous proposons dans ce chapitre de faire varier ce nombre pour constater son impact sur le temps de calcul et la qualité de la segmentation. En même temps nous comparons à chaque fois les résultats de la segmentation en utilisant les superpixels avec les résultats de la segmentation en utilisant tous les pixels.

III.2. Préambule

La technique de superpexilisation SLIC nécessite de lui fournir le nombre de superpixels à générer.

Rappelons que dans cette méthode les superpixels sont des petites régions homogènes et quasi-régulières. Il nous paraît évident que le nombre de superpixels choisi influera sur les résultats de la segmentation.

Nous noterons le nombre de superpixels dans l'image par K .

La figure III.1 fournit un exemple de superpexilisation avec des K différents.

Sur les figures III.1. b, c et d, nous affichons l'image original avec une délimitation en rouge des superpixels.

Sur les figures III.1. e, f et g nous affichons l'image des valeur moyenne des superpixels, c'est-à-dire chaque pixel d'un superpixel est remplacé par la valeur moyenne des pixels qu'il contient.

Lorsque ce nombre de superpixels est faible plusieurs pixels qui appartiennent à des classes différentes peuvent être regroupés dans la même région. Nous pouvons constater cela sur la figure III.1. En effet, si nous observons l'image de la figure III.1.b où le nombre de superpixels est 6 ($K=6$), nous pouvons constater par exemple que le superpixel numéroté 1 regroupe des pixels appartenant au ciel, au toit, à l'arbre et au mur.

Lorsque le nombre de superpixels augmente, comme dans le cas de la figure III.1.c où $K=10$, le nombre de pixels appartenant à des objets différents se trouve réduit dans le même superpixel. En effet le superpixel 1 contient principalement des pixels qui appartiennent au ciel, mais aussi quelques pixels qui appartiennent à l'arbre.

Pour un nombre de superpixels relativement grand comme dans le cas de la figure III.1.d où $K=100$, chaque superpixel regroupe essentiellement des pixels d'un même objet et les objets sont mieux délimités.



(a)



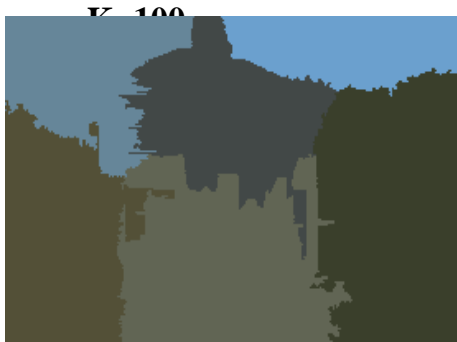
(b), $K=6$



(c), $K=10$



(d), $k=100$



(e)



Figure III.1: Exemple de superpexilisation d'une image
 (a) : image couleur original. (b) : image superpexilisées avec $K=6$.
 (c) : image superpexilisées pour $K=10$.
 (d) : image superpexilisées pour $K=100$. (e) image des valeurs moyennes de l'image (b).
 (f) : image des valeurs moyennes de l'image (d). (g) : image des valeurs moyennes de l'image (d).

III.3. Tests

Notre but est d'étudier l'influence de la superpexilisation sur la segmentation d'image, en utilisant l'algorithme k-means sur plusieurs images. Nous allons comparer les résultats de cette dernière avec les résultats de la segmentation en utilisant tous les pixels de l'image. Nous prélevons à chaque fois le temps de simulation pour chaque test.

Pour nous rendre compte de l'influence de la superpexilisation sur la segmentation d'image, nous proposons de tester 7 images dont les résultats sont donnés sur les figures III.2, III.3, III.4, III.5, III.6, III.7 et III.8.

Chaque figure donne les résultats de la segmentation pour une image.

Nous affichons sur la première colonne l'image originale, sur la deuxième colonne le résultat de la segmentation en utilisant l'algorithme k-means sur tous les pixels de l'image. Sur la troisième colonne nous affichons l'image des superpixels en valeur moyennes, c'est-à-dire tous les pixels d'un même superpixel est représentée par la valeur moyenne des pixels qui le constitue.

La dernière colonne fournie les résultats de la segmentation par k-means en utilisant seulement les superpixels.

Il est évident que le nombre de superpixels choisi K influera sur le résultat de la segmentation, nous proposons de tester plusieurs valeurs.

Comme les différentes images choisies sont de taille différentes nous proposons, au lieu de choisir un nombre de superpixels identique pour toutes les images, de travailler avec un taux de superpexilisation TS .

Nous réalisons une simple règle de trois en considérant que le nombre de pixels de l'image original correspond à un taux de 100% et que le nombre de superpixels correspond à un taux TS .

Nous avons choisi de tester 5 valeurs pour ce taux, soit 0.01%, 0.1%, 1%, 10% et 30%. Nous donnerons sur le tableau III.1 à chaque fois, le nombre de superpixels K qui correspond à chaque taux pour chaque image.

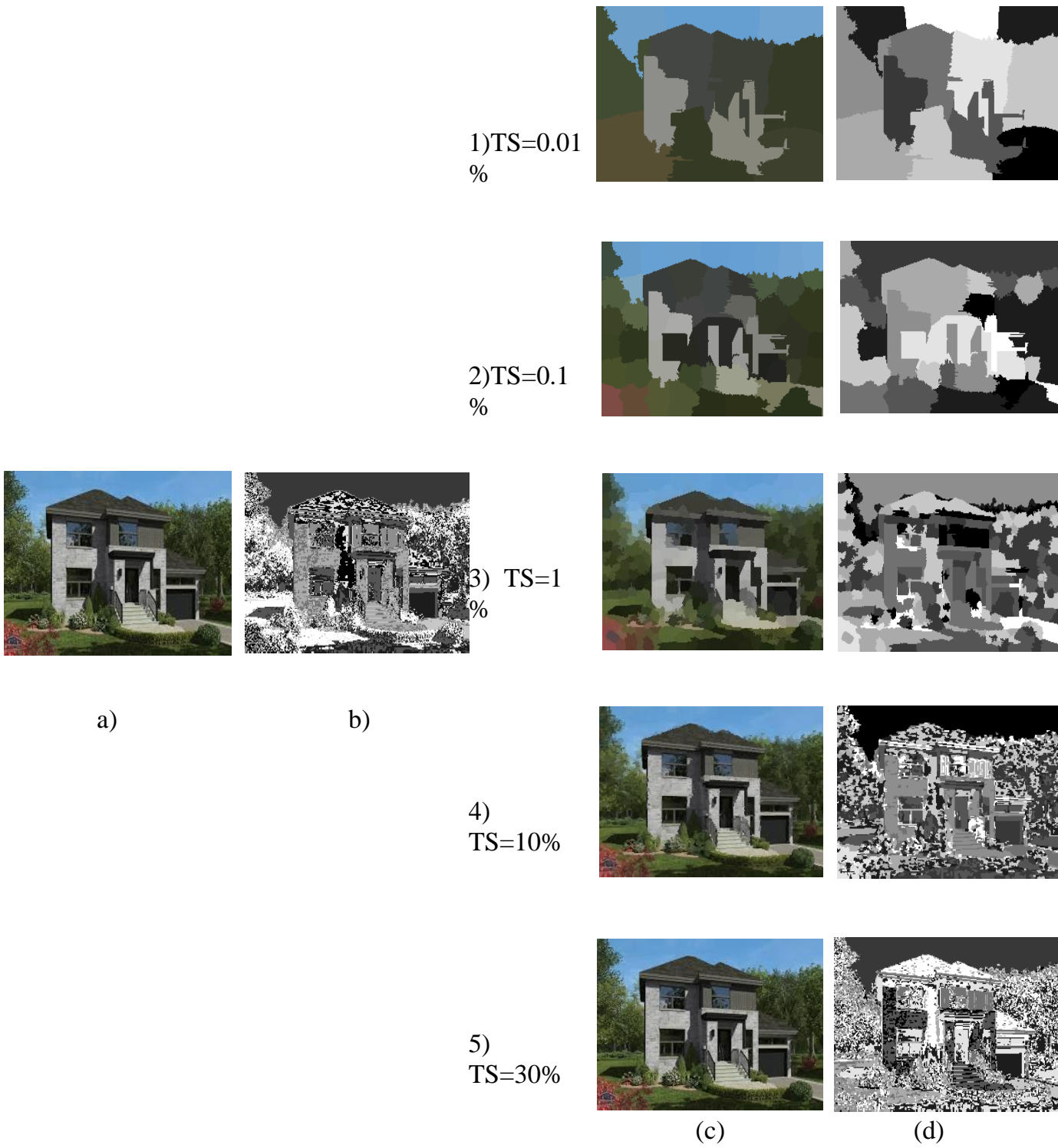


Figure III.2: résultat de la segmentation d'une image de taille 275x183 en faisant varier le Taux des pixels

- a) Image original
- b) Image segmentée par k-means
- c) Image des valeur moyenne
- d) Image des valeurs moyennes segmentées par k-means

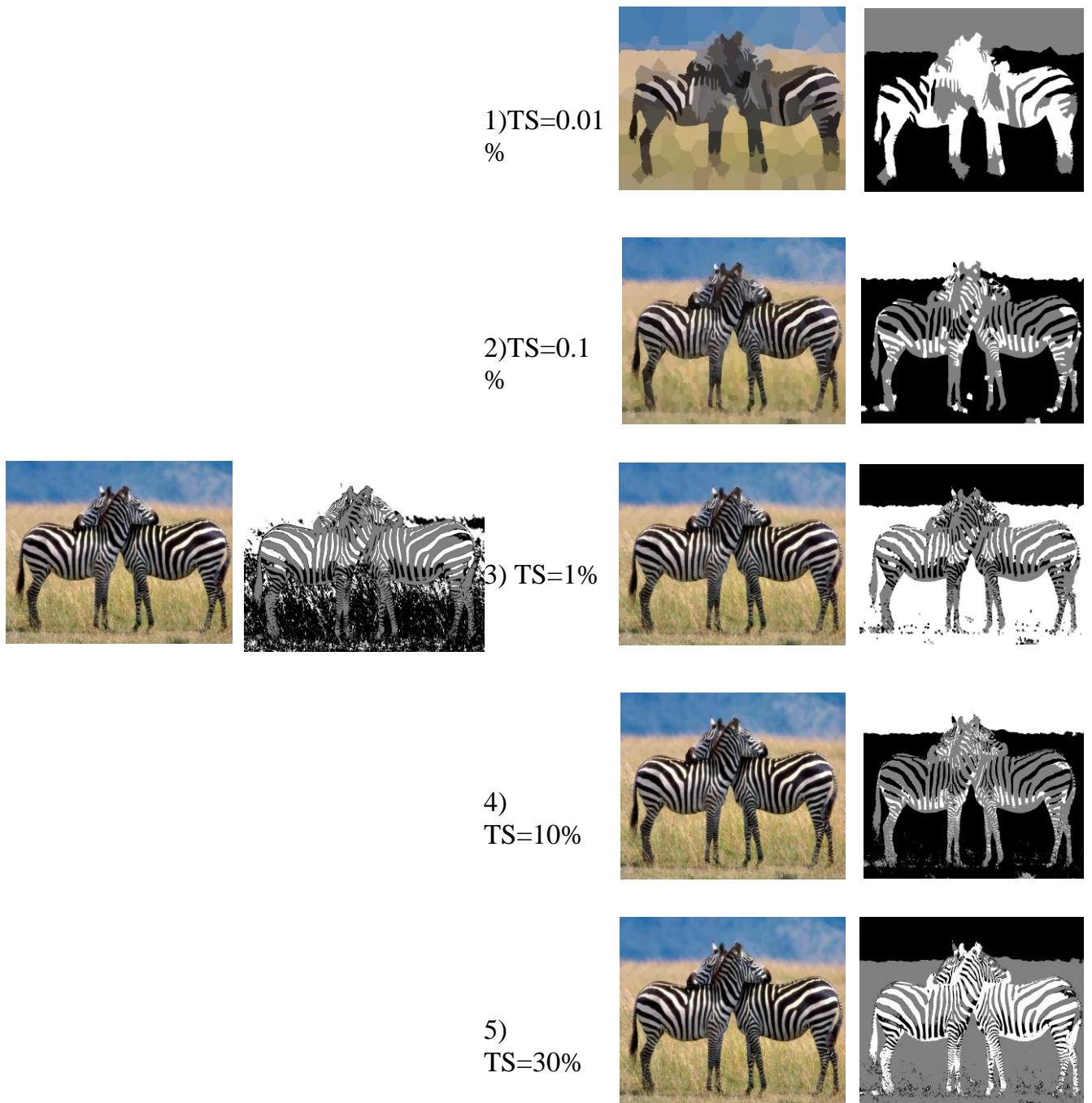


Figure III.3: résultat de la segmentation d'une image de taille 1600x1024 en faisant varier le Taux des pixels

- a) Image original
- b) Image segmentée par k-means
- c) Image des valeur moyenne

d) Image des valeurs moyennes segmentées par k-means

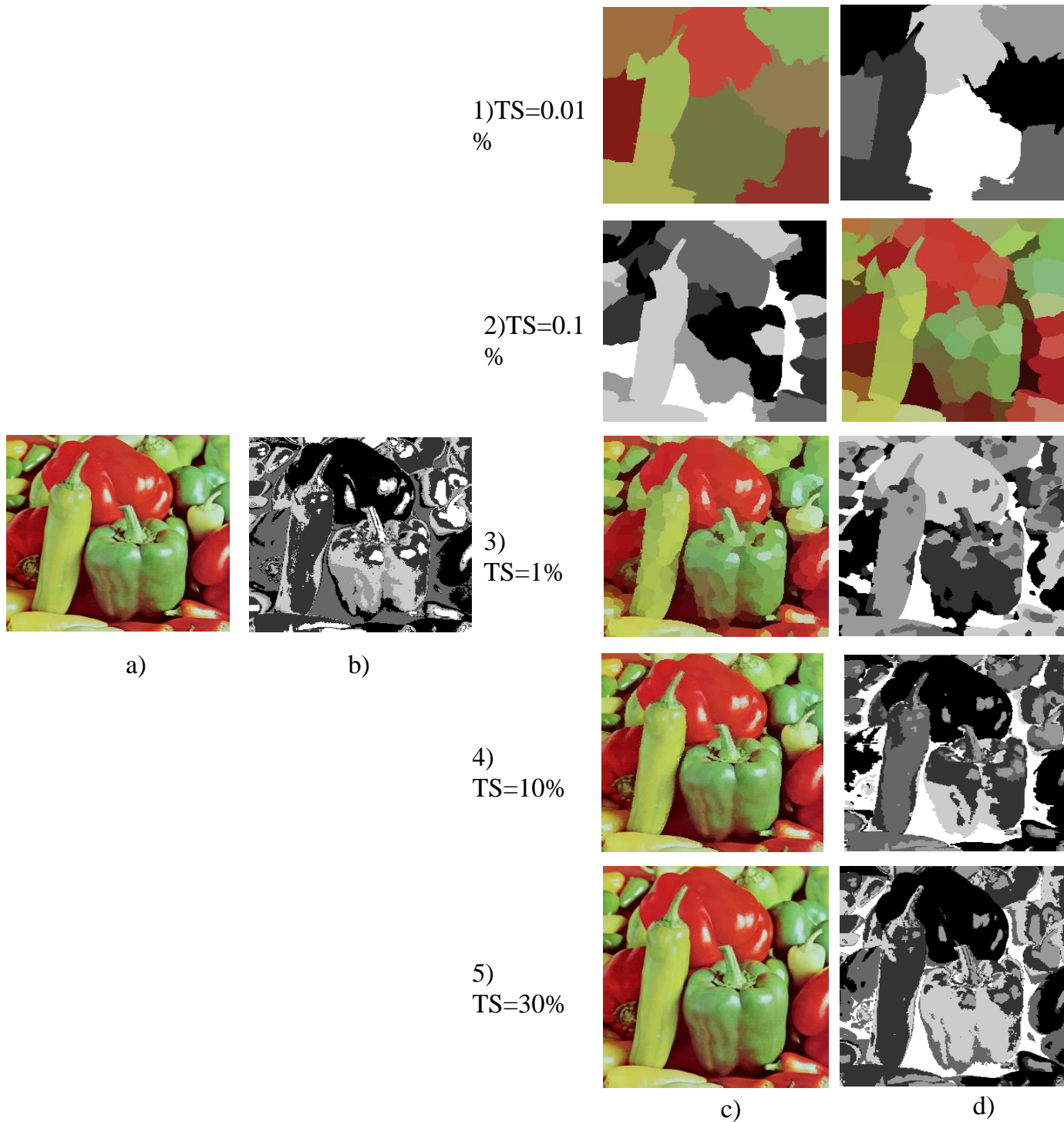


Figure III.4: résultat de la segmentation d'une image de taille 256x256 en faisant varier le Taux des pixels

- a) Image original
- b) Image segmentée par k-means
- c) Image des valeur moyenne

d) Image des valeurs moyennes segmentées par k-means

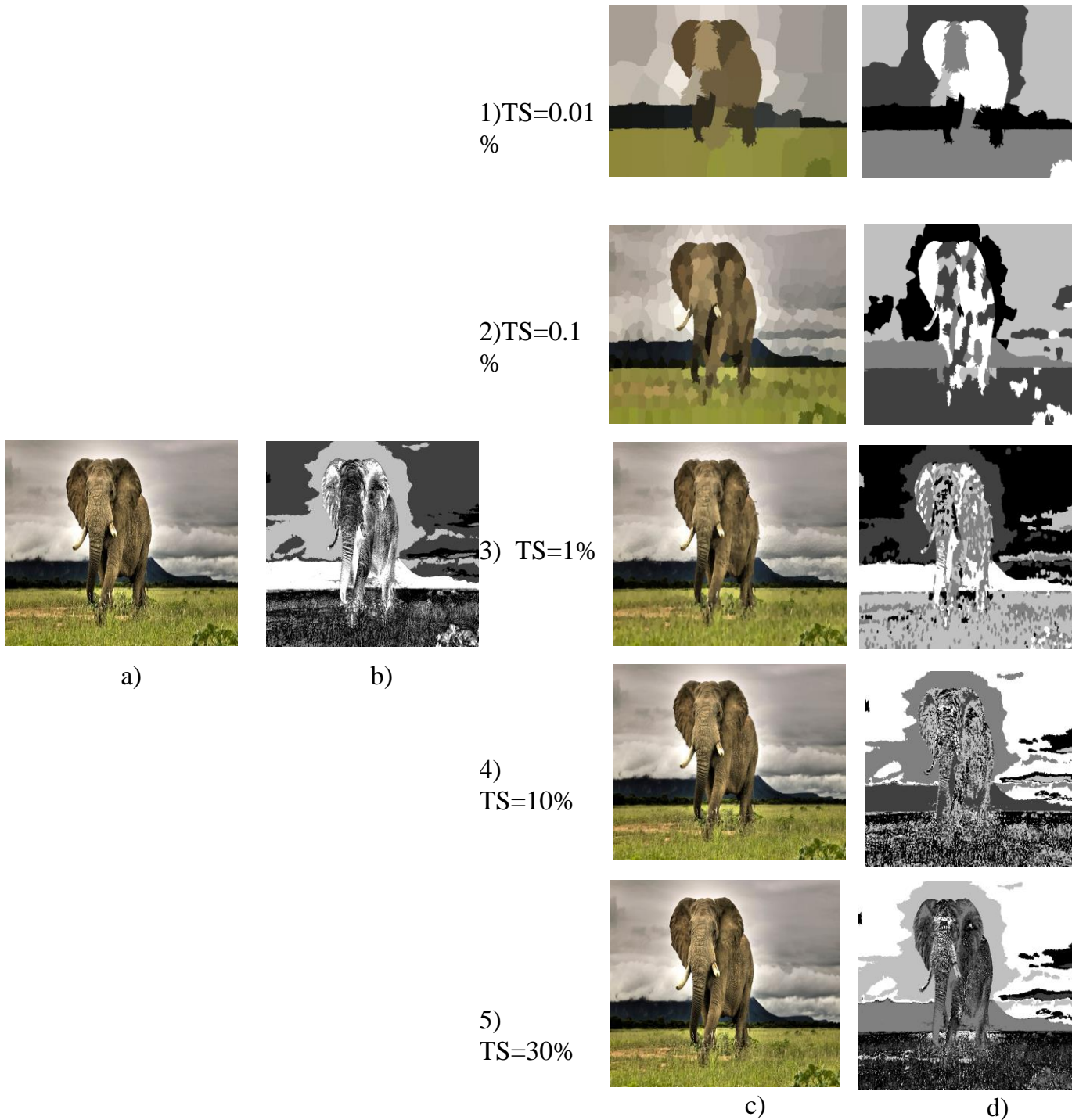


Figure III.5: résultat de la segmentation d'une image de taille 1000x500 en faisant varier le Taux des pixels

- a) Image original
- b) Image segmentée par k-means
- c) Image des valeur moyenne
- d) Image des valeurs moyennes segmentées par k-means

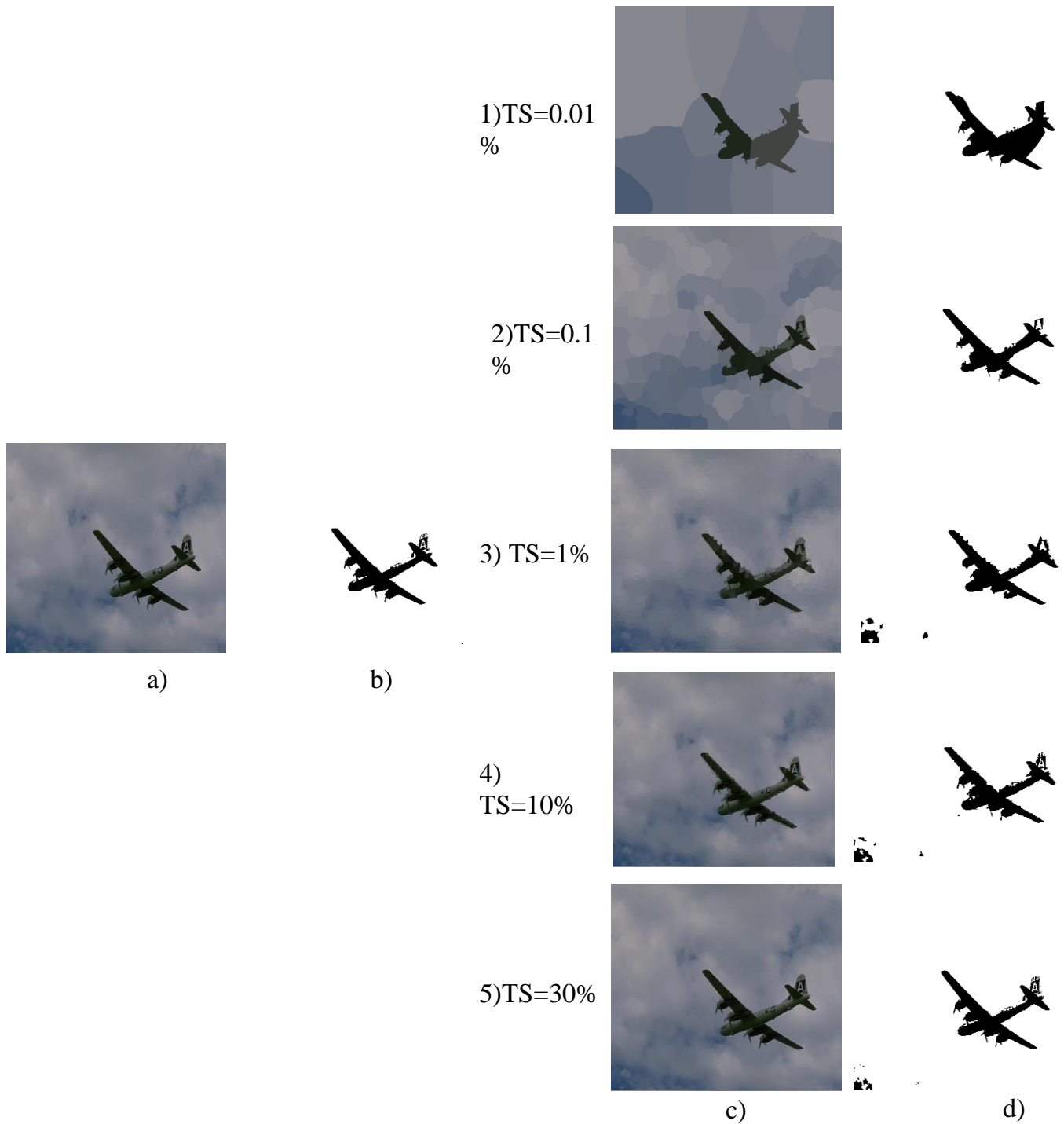


Figure III.6: résultat de la segmentation d'une image de taille 481x321 en faisant varier le Taux des pixels

- a) Image original
- b) Image segmentée par k-means
- c) Image des valeur moyenne
- d) Image des valeurs moyennes segmentées par k-means

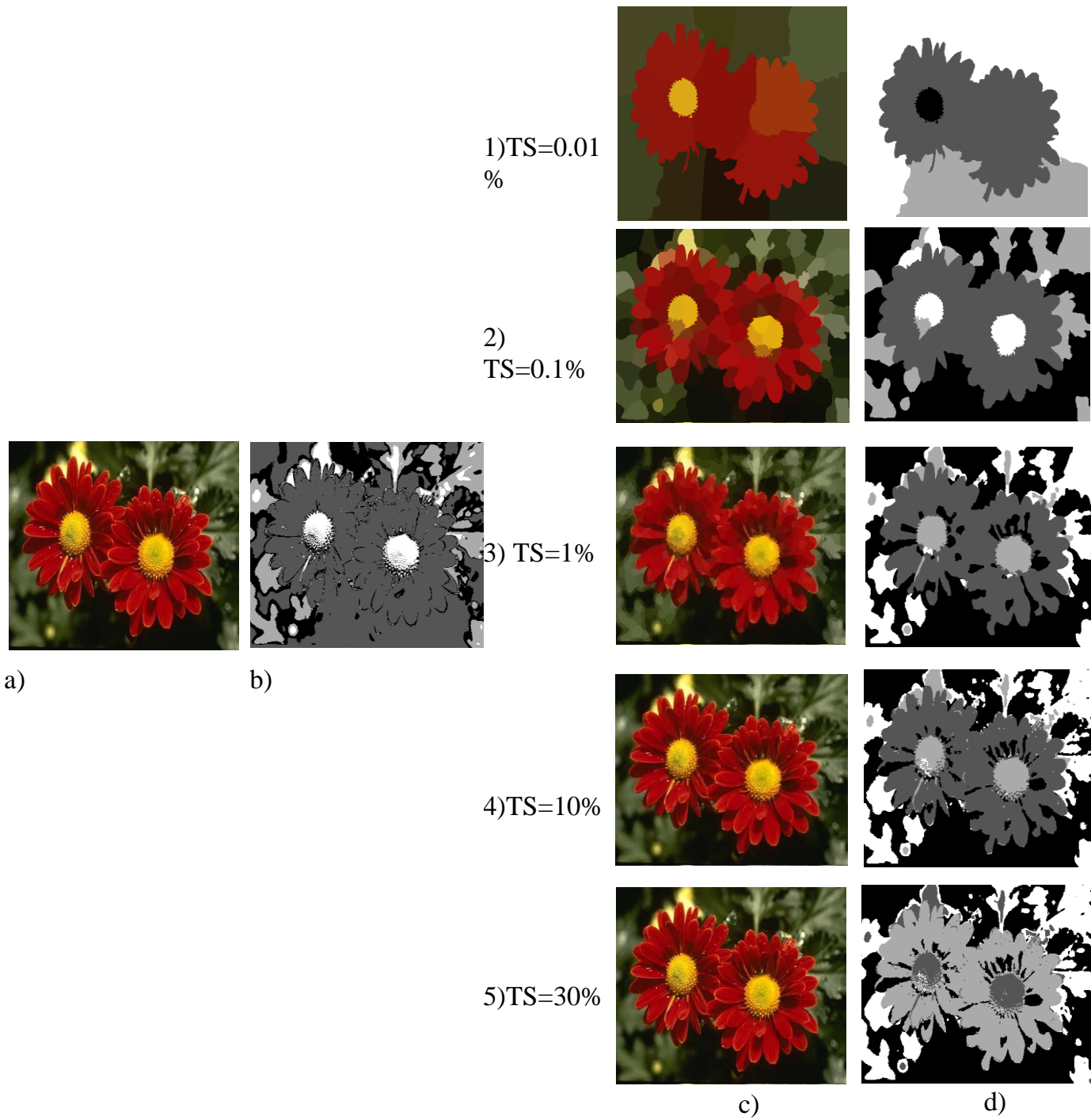


Figure III.7: résultat de la segmentation d'une image de taille 256x256 en faisant varier le Taux des pixels

- a) Image original
- b) Image segmentée par k-means
- c) Image des valeur moyenne
- d) Image des valeurs moyennes segmentées par k-means

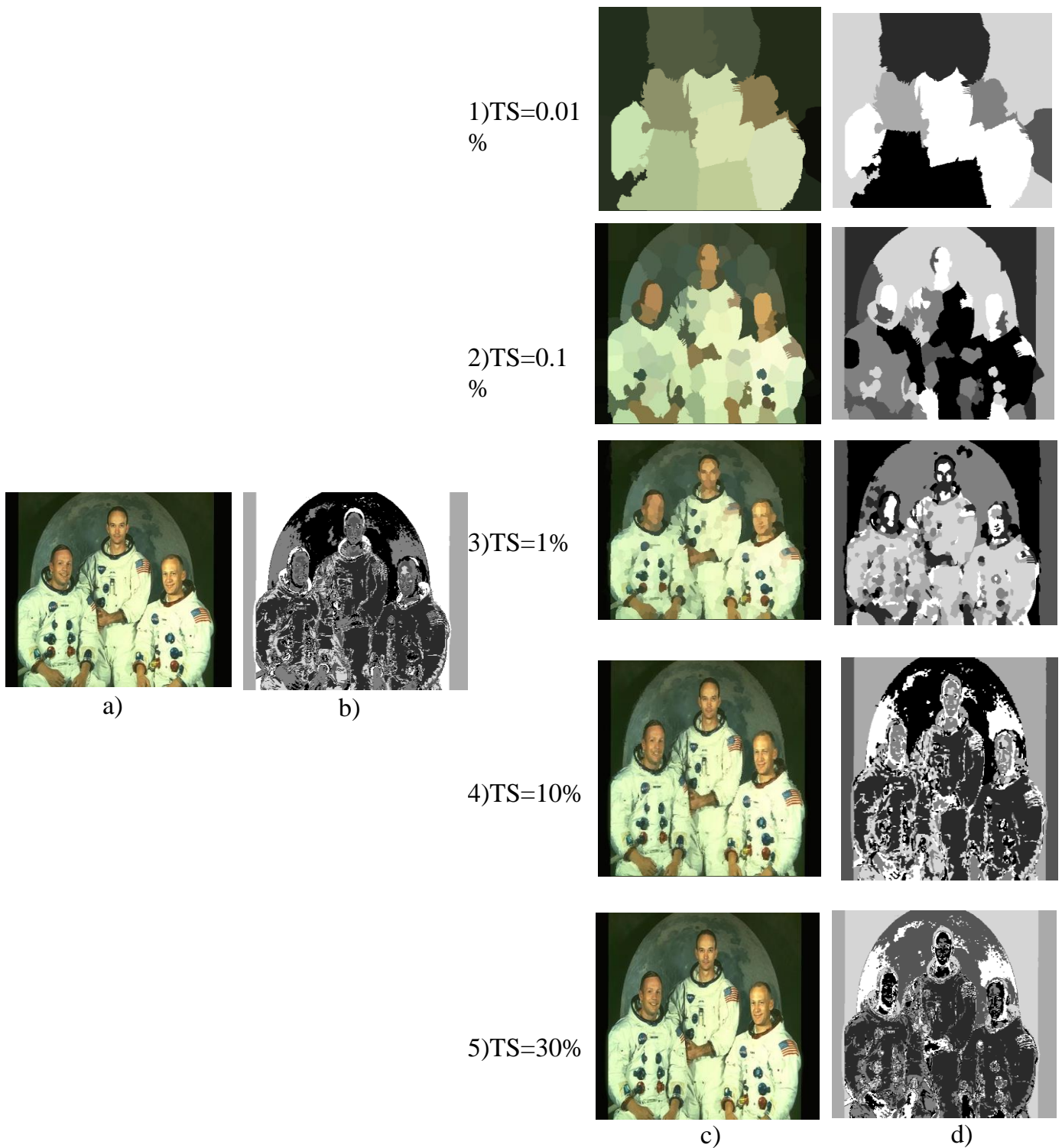


Figure III.8: résultat de la segmentation d'une image de taille 481x321 en faisant varier le taux de superpexilisation

- a) Image original
- b) Image segmentée par k-means
- c) Image des valeur moyenne
- d) Image des valeurs moyennes segmentées par k-means

image	taille	Nbre-classes	TS	K	Temps (s) K-means « images superpixelisées »	Temps (s) k-means « images originaux »
maison	275x183	10	0.01 %	15	1.652219	56.421582
			0.1 %	50	1.771396	
			1 %	503	1.896994	
			10 %	5032	7.208833	
			30%	15097	31.132911	
zèbre	1600x1024	3	0.01 %	163	4.083243	731.727520
			0.1 %	1638	4.634247	
			1 %	16384	10.100812	
			10 %	163840	84.914453	
			30%	491520	509.958049	
peppers	256x256	6	0.01 %	7	1.217068	82.089017
			0.1 %	65	1.267232	
			1 %	655	1.770333	
			10 %	6553	6.006521	
			30%	19661	20.919042	
éléphant	1000x500	5	0.01 %	50	1.731099	413.045613
			0.1 %	500	2.129589	
			1 %	5000	4.615721	
			10 %	50000	42.501380	
			30%	150000	243.576652	
avion	481x321	2	0.01 %	15	1.163801	18.040057
			0.1 %	154	1.295677	
			1 %	1544	1.743727	
			10 %	15440	5.810216	
			30%	46320	13.9975445	
fleurs	481x321	4	0.01 %	15	1.336061	97.585808
			0.1 %	154	1.468756	
			1 %	1544	2.331986	
			10 %	15440	7.238899	
			30%	46320	24.206389	
astronaute	481x321	7	0.01 %	15	1.414983	163.774553
			0.1 %	154	1.551977	
			1 %	1544	2.343710	
			10 %	15440	11.742798	
			30%	46320	80.053752	

III.1. Tableau 1 : Résultats de l'application de k-means sur les différentes images

III.4 Interprétation

Nous remarquons que lorsque le taux de superpexilisation est faible les résultats de la segmentation sont mauvais. En effet lorsque $TS = 0.01\%$ nous constatons par exemple sur la figure III.2.d.1 que les résultats de la segmentation sont mauvais. Par exemple deux superpixels qui représentent le ciel (en haut à droite et à gauche) et un superpixel qui représente le sol (en bas à droite) ont été classés dans la même classe.

C'est la même chose avec les autres images par exemple dans la Figure III.8.d.1 les têtes des astronautes ne sont pas bien segmentées, en effet par exemple l'astronaute du milieu sa tête est confondue avec une partie de la lune. Ce résultat est attendu car le nombre de superpixels est très faible. Dans ce cas pour ces images. Il est de $K = 15$ pour l'image maison et de $K = 15$ pour l'image astronaute (voir tableau III.1).

Lorsque le taux de superpexilisation TS augmente, par conséquent le nombre de superpixels NS augmente, le résultat de la segmentation s'améliore.

Toutes les figures (III.2, III.3, III.4, III.5, III.6, III.7 et III.8) montrent clairement qu'à partir d'un $TS = 10\%$ les résultats de la segmentation sont assez bon comparativement aux résultats obtenus en utilisant tous les pixels des images.

Nous remarquons, qu'à partir de ce Taux (10%) l'image des valeurs moyennes des superpixels est visuellement assez semblables à l'image réelle.

Nous pouvons dire donc que le Taux 10% peut représenter l'image réelle.

Les temps de calculs sont nettement diminués en utilisant la superpexilisation comme le montre le tableau III.1.

Il est bien sur relativement très faible pour les Taux 0.01% et 0.1%. Il reste faible pour un Taux de 10%.

Il passe par exemple pour l'image éléphant de 413s dans le cas de la segmentation en utilisant tous les pixels à 42s en utilisant la superpexilisation avec un Taux de 10%.

Il faut noter que lors du calcul de ce temps nous avons inclus le temps de superpexilisation, le temps de la segmentation est donc plus faible. Il est donc dans le cas de l'image éléphant inférieur à 42s.

Nous considérons que le gain dans le temps de calcul est conséquent pour une faible perte d'information en utilisant un Taux (TS=10%).

III.5 Conclusion

Dans ce chapitre nous avons présenté les différents résultats obtenus par la méthode de superpexilisation décrit dans les chapitres précédents.

Pour mieux situer cette dernière, des tests ont été effectués dans le but de comparer son influence sur les résultats des autres méthodes de segmentations, La méthode que nous avons choisie est celle de k-means.

Dans un premier temps nous l'avons appliqué pour tous les pixels de l'image, puis nous l'avons appliqué sur les mêmes images, mais qui ont été traitées par l'algorithme SLIC (Simple Linéaire Itératif Clustering).

D'après notre expérimentation on peut constater que l'application de l'algorithme de k-means sur des images superpexilisées est parfait en terme de temps de calcul, comme il est possible de dire à travers ces résultats que le fait d'augmenter le taux de superpixels TS les résultats obtenus sont meilleurs d'un point de vue visuel.

La segmentation d'images est une étape importante car elle conditionne la qualité des opérations ultérieures dans processus de traitement d'images.

Dans ce travail, nous avons étudié la segmentation d'images et ses différentes approches.

Comme cette opération peut prendre un temps de calcul important surtout pour des images à grandes dimensions, nous avons proposé de réduire ce temps en effectuant une superpexilisation avant la segmentation.

Après avoir étudié la segmentation d'image et ses différentes approches, nous avons proposé une méthode qui se base particulièrement sur la superpexilisation.

Cette dernière consiste à regrouper les pixels de l'image en petites zones homogènes réduisant ainsi le nombre d'éléments à traiter, et par conséquent le temps de calcul.

Il existe plusieurs méthodes de superpexilisation, dans ce travail nous avons choisi d'étudier la méthode SLIC (Simple Linéaire Itératif Clustering).

La segmentation est par la suite réalisée en utilisant l'algorithme k-means.

Pour évaluer les résultats obtenus, nous les avons comparés avec ceux obtenus en réalisant la segmentation sur les images brutes c'est-à-dire sans superpexilisation.

Les résultats nous paraissent promoteurs car les temps de calcul sont réduits sans pour autant perdre en qualité de la segmentation.

Il sera utile d'évaluer les résultats de manière objective en utilisant par exemple des images synthétiques.

Il existe d'autres méthodes de superpexilisation qui peuvent être utilisées à la place de SLIC comme SCALP (Superpixels with Contour Adhérence using Linear Path), ou ETPS (Extended Topology Preserving Superpixels).

Il sera judicieux de les tester pour vérifier si elles donnent de meilleurs résultats.

Il est utile de noter que l'algorithme k-means est très performant en terme de temps d'exécution, mais il souffre de problème de dépendance des résultats aux choix effectués lors de l'initialisation.

D'autres algorithmes de classifications peuvent aussi être utilisés.

Bibliographie

- [1] : J.P COCQUEREZ et S PHILIPP – Analyse d’images : filtrage et segmentation, Université Masson Paris – 1995.
- [2] : Fatima EDDAOUDI, aide à la détection par les techniques de segmentation et de codage de textures des images, faculté des sciences université Mohammed v – agdal.
- [3] : Hayet MEDROUK et Souad TOUCHERIFT, segmentation par classification d’une partie de l’image, mémoire de fin d’étude, ummto, 2010.
- [4] : Adel BELAID , Kamel YAHIAOUI, un système basé sur la vision par ordinateur pour la prévention des dangers des chameaux dans les routes sahariennes, mémoire master académique, université kasdi merbah – Ouargla, 2016.
- [5] : Z.GUELLIL et L.ZAOUI, proposition d'une solution au problème d’initialisation cas du k-means, université des sciences et de la technologie d’Oran MB, université Mohamed Boudiaf usto -BP 1505 el mnaouer -Oran – Algérie.
- [6] : Mounzer BOUBOU, contribution aux méthode de classification non supervisée via des approches prétopologiques et d’agrégation d’opinions, thèse de doctorat août 2006.
- [7] : Radhakrishna ACHANTA, Kevin SMITH, AURELIEN LUCCHI and Sabine SUSSTRUNK, SLIC Superpixels, School of Computer and Communication Sciences (IC), Ecole Polytechnique Fédérale de Lausanne (EPFL).
- [8] : Remi GIRAUD, Vinh-Thong Ta, Nicolas PAPADAKIS. Décomposition en superpixels via l’utilisation de chemin linéaire. GRETSI, Juan-les-Pins, France. Sep 2017.
- [9] : Remi GIRAUD. Algorithmes de correspondance et superpixels pour l’analyse et le traitement d’images. Autre [cs.OH]. Université de Bordeaux, 2017.
- [10]: Remi GIRAUD, vinh-thong TA, Nicolas PAPADAKIS. Evaluation Framework of Superpixel Methods with a Global Regularity Measure. Journal of Electronic Imaging, Society of Photo-optical Instrumentation Engineers, 2017.
- [11]: Carl YUHENG REN, IAN REID, gSLIC: a real-time implementation of SLIC superpixel segmentation, University of Oxford, Department of Engineering Parks Road, Oxford, UK, June 28, 2011.
- [12]: Mme Marie-anne BAUDA, compréhension de scènes urbaines par combinaison d’information 2d/3d, Institut National Polytechnique de Toulouse (INP Toulouse) Discipline ou spécialité : Image, Information et Hypermédiale, lundi 13 juin 2016
-