



Institut de Recherche  
en Informatique de Toulouse  
CNRS - INP - UT3 - UT1 - UT2J

# Rapport Technique sur l'Implémentation des Algorithmes de Segmentation d'Image

EL HEYBA El Heyba

August 2, 2024

# Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Dépendances et Compilation</b>	<b>3</b>
<b>3</b>	<b>Exécution</b>	<b>3</b>
<b>4</b>	<b>Structure du Code</b>	<b>3</b>
<b>5</b>	<b>Détails des Algorithmes</b>	<b>4</b>
5.1	K-means . . . . .	4
5.1.1	Fonctions Auxiliaires . . . . .	4
5.2	Méthode d'Évaluation . . . . .	5
5.2.1	Matrice de Confusion . . . . .	5
5.2.2	Précision et Sensibilité . . . . .	5
5.2.3	Coefficient de Similarité de Dice (DSC) . . . . .	5
<b>6</b>	<b>Interprétation des Résultats</b>	<b>6</b>
6.1	Mean-Shift . . . . .	6
6.1.1	Fonctions Auxiliaires . . . . .	6
6.2	Supapixel . . . . .	8
6.2.1	Fonctions Auxiliaires . . . . .	8
<b>7</b>	<b>Résultats et Visualisation</b>	<b>9</b>
<b>8</b>	<b>Références</b>	<b>9</b>

# 1 Introduction

Ce projet implémente plusieurs techniques de segmentation d'image en utilisant la bibliothèque OpenCV. Les algorithmes implémentés incluent K-means, Mean-Shift, ainsi qu'une méthode de superpixels basée sur K-means dans l'espace de couleur Lab. L'objectif est de segmenter les images en regroupant les pixels ayant des caractéristiques similaires.

## 2 Dépendances et Compilation

Le code utilise OpenCV et OpenMP pour le traitement d'image et la parallélisation. Assurez-vous que ces bibliothèques sont installées avant de compiler le code.

- **Dépendances :**
  - OpenCV
  - OpenMP (pour la parallélisation)

## 3 Exécution

Pour exécuter le programme après compilation, utilisez la commande suivante :

```
1 ./image_segmentation <image_file> <K_num_of_clusters> [<  
  image_ground_truth>]
```

Listing 1: Exécution du programme

```
1 ./image_segmentation "../images/texture8.png" 2 "../  
  images/texture8_VT.png"
```

Listing 2: Exemple d'exécution du programme

- **<image\_file>** : Chemin vers l'image à segmenter.
- **<K\_num\_of\_clusters>** : Nombre de clusters pour l'algorithme K-means.
- **<image\_ground\_truth>** (optionnel) : Chemin vers l'image de vérité terrain pour évaluation.

## 4 Structure du Code

Le code est structuré autour des fonctions principales suivantes :

- **K-means** : Segmente l'image en  $k$  clusters.
- **Mean-Shift** : Une technique non paramétrique pour trouver les modes de densité dans l'espace des caractéristiques.
- **Superpixel** : Segmentation de l'image en superpixels en utilisant K-means dans l'espace de couleur Lab.

## 5 Détails des Algorithmes

### 5.1 K-means

#### Fonction : `kmeans`

Cette fonction implémente l'algorithme K-means pour segmenter une image en  $k$  clusters.

- **Initialisation** : Les centres de clusters sont initialisés à partir des vecteurs d'entrée.
- **Boucle Principale** :
  - Calcul de la distance euclidienne entre chaque pixel et les centres.
  - Attribution de chaque pixel au centre le plus proche.
  - Mise à jour des centres en calculant la moyenne des pixels attribués.

#### 5.1.1 Fonctions Auxiliaires

Pour le calcul des distances et la détermination du centre le plus proche, les fonctions auxiliaires suivantes sont utilisées :

- **Fonction de calcul de la distance euclidienne** : Calcule la distance euclidienne entre deux couleurs en espace RGB en utilisant les valeurs des composantes rouge, verte et bleue. Elle retourne la distance entre les deux couleurs.
- **Fonction de recherche du centre le plus proche** : Détermine l'indice du centre de cluster le plus proche pour une couleur donnée en calculant la distance euclidienne entre la couleur du pixel et tous les centres de clusters disponibles. Retourne l'indice du centre le plus proche.

L'algorithme s'arrête lorsque les centres ne changent plus ou après un nombre maximal d'itérations.

## 5.2 Méthode d'Évaluation

La segmentation d'image est une étape cruciale dans de nombreuses applications de vision par ordinateur, où l'objectif est de partitionner une image en régions significatives. L'algorithme k-means est l'une des méthodes les plus populaires pour effectuer cette tâche. Ce rapport présente l'implémentation de l'algorithme k-means pour la segmentation d'images et l'évaluation de la qualité de la segmentation.

Pour évaluer la qualité de la segmentation, nous comparons les résultats obtenus par l'algorithme k-means avec une vérité terrain (ground truth) si disponible. Les métriques utilisées pour cette évaluation sont les suivantes :

### 5.2.1 Matrice de Confusion

Pour chaque classe  $c$ , nous définissons une matrice de confusion qui résume les performances de la classification. Cette matrice est structurée comme suit :

	Prédit $c$	Prédit $\neg c$
Réel $c$	<b>TP (True Positives)</b>	<b>FN (False Negatives)</b>
Réel $\neg c$	<b>FP (False Positives)</b>	<b>TN (True Negatives)</b>

- **TP (True Positives)** : Nombre de pixels correctement classés dans la classe  $c$ .
- **FP (False Positives)** : Nombre de pixels incorrectement classés dans la classe  $c$ .
- **TN (True Negatives)** : Nombre de pixels correctement classés dans les classes autres que  $c$ .
- **FN (False Negatives)** : Nombre de pixels incorrectement classés dans les classes autres que  $c$ .

### 5.2.2 Précision et Sensibilité

$$\text{Précision} = \frac{TP}{TP + FP} \quad (1)$$

$$\text{Sensibilité} = \frac{TP}{TP + FN} \quad (2)$$

### 5.2.3 Coefficient de Similarité de Dice (DSC)

Le coefficient de Dice mesure la similarité entre deux ensembles :

$$\text{DSC} = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \quad (3)$$

## 6 Interprétation des Résultats

Les résultats de l'évaluation fournissent une vue d'ensemble de la performance de la segmentation pour chaque classe. Une précision et une sensibilité élevées indiquent une bonne performance. Le coefficient de Dice, étant une mesure de similarité, varie entre 0 (aucune similarité) et 1 (similarité parfaite).

L'évaluation de la qualité de la segmentation est essentielle pour améliorer les algorithmes et obtenir des résultats plus précis. L'algorithme k-means, bien que simple, peut fournir des résultats satisfaisants, mais son efficacité dépend de la bonne détermination du nombre de clusters  $k$ . Les métriques de qualité telles que la précision, la sensibilité et le coefficient de Dice offrent des outils quantitatifs pour comparer et évaluer les performances des algorithmes de segmentation.

### 6.1 Mean-Shift

**Fonction : meanshift**

L'algorithme Mean-Shift est utilisé pour trouver les modes de densité dans l'image. Il fonctionne en déplaçant les points de données vers la densité de données la plus élevée jusqu'à ce que la convergence soit atteinte.

- **Initialisation** : Matrices pour stocker les moyennes et les comptages des pixels.
- **Boucle Principale** :
  - Pour chaque pixel, calcul des distances chromatiques avec les pixels voisins.
  - Mise à jour des moyennes et des comptages pour chaque pixel.
  - Mise à jour des positions des pixels en fonction des moyennes calculées.

#### 6.1.1 Fonctions Auxiliaires

Pour le calcul des distances chromatiques et la vérification de la convergence, les fonctions auxiliaires suivantes sont utilisées :

- **Comparateur personnalisé pour Vec3f** : Structure permettant de comparer deux vecteurs à trois composantes (Vec3f) en utilisant un ordre lexicographique. Elle est utilisée dans une carte pour stocker les modes détectés.
- **Fonction de calcul de la distance chromatique** : Calcule la distance chromatique entre deux pixels en utilisant la norme euclidienne. Si la distance est inférieure à un seuil chromatique donné, elle met à jour les matrices des moyennes et des comptages des pixels concernés.
- **Fonction de vérification de la convergence** : Vérifie si la norme euclidienne de tous les éléments de la matrice de différence entre deux itérations successives est inférieure à un seuil de tolérance. Retourne vrai si la différence est supérieure au seuil, indiquant que l'algorithme n'a pas encore convergé.

L'algorithme s'arrête lorsque la différence entre les itérations successives est inférieure à un certain seuil ou après un nombre maximal d'itérations.

**Fonction : detectModes**

La fonction **detectModes** est conçue pour détecter les modes dans une image segmentée. Elle utilise une carte pour stocker les modes détectés et leur comptage. La détection est parallélisée pour améliorer les performances. L'algorithme de détection des modes est décrit par le pseudo-code suivant :

---

**Algorithm 1** Détection des Modes

---

**Require:** Matrice de l'image segmentée  $matIMG$ , tolérance  $\epsilon$

**Ensure:** Liste des modes détectés

```
1: Initialiser une carte  $modeMap$  pour stocker les modes et leur comptage
2: Initialiser un vecteur  $modes$  pour stocker les modes
3: for chaque pixel  $pixel$  dans  $matIMG$  do
4:    $minDist \leftarrow \infty$ 
5:    $closestMode \leftarrow \text{null}$ 
6:   for chaque mode  $mode$  dans  $modes$  do
7:      $dist \leftarrow \|mode - pixel\|$ 
8:     if  $dist < minDist$  then
9:        $minDist \leftarrow dist$ 
10:       $closestMode \leftarrow mode$ 
11:    end if
12:  end for
13:  if  $minDist < \epsilon$  then
14:    Incrémenter le comptage de  $closestMode$  dans  $modeMap$ 
15:  else
16:    Ajouter  $pixel$  comme un nouveau mode dans  $modes$ 
17:    Initialiser le comptage de  $pixel$  dans  $modeMap$  à 1
18:  end if
19: end for
20: Trier les modes dans  $modes$  par leur comptage dans  $modeMap$  (optionnel)
21: return  $modes = 0$ 
```

---

## 6.2 Superpixel

**Fonction :** `superpixel`

Cette fonction utilise l'algorithme K-means dans l'espace de couleur Lab pour segmenter l'image en superpixels.

- **Prétraitement** : Conversion de l'image en espace de couleur Lab.
- **Initialisation** : Création d'un vecteur de caractéristiques comprenant les coordonnées spatiales et les valeurs de couleur Lab.
- **K-means** : Clustering des vecteurs de caractéristiques en  $k$  clusters.
- **Post-traitement** : Conversion des centres en espace RGB pour l'affichage des résultats.

### 6.2.1 Fonctions Auxiliaires

Pour le calcul des distances et la détermination des centres les plus proches, les fonctions auxiliaires suivantes sont utilisées :



- **Fonction de calcul de la distance euclidienne entre deux vecteurs à 5 canaux** : Calcule la distance euclidienne entre deux vecteurs à cinq canaux, représentant les coordonnées spatiales et les valeurs de couleur Lab. Elle mesure la similarité entre les vecteurs de caractéristiques des pixels et les centres de clusters.
- **Fonction de détermination du centre le plus proche** : Détermine l'indice du centre de cluster le plus proche pour un vecteur de caractéristiques donné. Elle utilise la fonction de calcul de distance euclidienne pour comparer le vecteur avec tous les centres de clusters disponibles et trouve celui avec la distance minimale.
- **Fonction de calcul des centres avec K-means** : Implémente l'algorithme K-means pour segmenter les vecteurs de caractéristiques en  $k$  clusters. Elle initialise les centres de clusters, attribue chaque vecteur de caractéristiques au centre le plus proche, met à jour les centres en calculant la moyenne des vecteurs attribués, et itère jusqu'à la convergence.

## 7 Résultats et Visualisation

Le programme affiche les résultats de la segmentation après chaque itération des algorithmes, permettant une visualisation dynamique du processus de convergence. Les résultats finaux montrent les images segmentées en fonction des différents algorithmes appliqués.

## 8 Références

- Documentation OpenCV
- Polycopié de cours par Sylvie Chambon
- Guides de Référence OpenMP