

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

EL HEYBA El Heyba

July 4, 2024

1 Introduction

Ce projet implémente plusieurs techniques de segmentation d'image en utilisant la bibliothèque OpenCV. Les algorithmes implémentés incluent K-means et Mean-Shift, ainsi qu'une méthode de superpixel basée sur K-means dans un 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 l'affichage des résultats pour la parallélisation. Assurez-vous de les avoir installés avant de compiler le code.

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

3 Exécution

Pour exécuter le programme après compilation :

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

Listing 1: Exécution du programme

```
1 ./main "../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 de plusieurs fonctions principales :

- **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.
- **Supapixel** : 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 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 comme la moyenne des pixels attribués.

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

5.2 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.

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 ϵ

Ensure: Liste des modes détectés

```
1: Initialiser un map  $modeMap$  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:    $found \leftarrow false$ 
5:   for chaque mode  $mode$  dans  $modes$  do
6:     if  $\|mode - pixel\| < \epsilon$  then
7:       Incréments le comptage de  $mode$  dans  $modeMap$ 
8:        $found \leftarrow true$ 
9:       break
10:    end if
11:  end for
12:  if non trouvé then
13:    Ajouter  $pixel$  comme un nouveau mode dans  $modes$ 
14:    Initialiser le comptage de  $pixel$  dans  $modeMap$  à 1
15:  end if
16: end for
17: Trier les modes dans  $modes$  par leur comptage dans  $modeMap$  (optionnel)
18: return  $modes$ 
```

5.3 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 BGR pour l'affichage des résultats.

6 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.

7 Conclusion

soon

8 Références

- OpenCV Documentation
- Polycopié de cours par Sylvie Chambon
- OpenCV Reference Guides