



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

Guide sur l'Implémentation de la main

EL HEYBA El Heyba

August 2, 2024

Contents

1	Introduction	2
2	Fonctions et Paramètres	2
2.1	printHelp	2
2.2	superpixel	2
3	Fonction principale main	3
4	Choix de Conception	5
5	Conclusion	5

1 Introduction

Ce document présente un rapport technique sur l'implémentation de la segmentation d'images en utilisant l'algorithme des superpixels en C++. Le code utilise la bibliothèque OpenCV pour la manipulation et le traitement des images. Le programme segmente une image en superpixels en utilisant l'algorithme SLICO (Simple Linear Iterative Clustering with zero parameter tuning).

2 Fonctions et Paramètres

2.1 printHelp

Cette fonction affiche un message d'aide pour l'utilisation du programme.

```
1 void printHelp(const string& progName) {  
2     cout << "Usage: " << progName << " <image_file> <  
    K_num_of_clusters> [<image_ground_truth>]" << endl  
    ;  
3 }
```

Listing 1: Définition de la fonction printHelp

Paramètre:

- progName: Le nom du programme.

2.2 superpixel

Cette fonction segmente une image en superpixels en utilisant l'algorithme SLICO.

```
1 Mat superpixel(const Mat img, int k, int m = 10, int  
    maxIter = 200) {  
2     Mat convertedImg;  
3     cvtColor(img, convertedImg, COLOR_BGR2Lab);
```

```

4
5     Ptr<ximgproc::SuperpixelSLIC> slic = ximgproc::
        createSuperpixelSLIC(convertedImg, ximgproc::SLICO
            , k, (float)m);
6     slic->iterate(maxIter);
7
8     Mat labels;
9     slic->getLabels(labels);
10
11    Mat mask, result;
12    slic->getLabelContourMask(mask, true);
13    img.copyTo(result, mask);
14
15    return result;
16 }

```

Listing 2: Définition de la fonction `superpixel`

Paramètres:

- `img`: L'image d'entrée à segmenter.
- `k`: Le nombre de superpixels.
- `m`: Le paramètre compactness (par défaut 10).
- `maxIter`: Le nombre maximal d'itérations (par défaut 200).

Description:

- La fonction convertit d'abord l'image en espace de couleur Lab.
- Ensuite, elle crée un objet SLICO pour la segmentation.
- Elle exécute l'algorithme d'itération pour générer les superpixels.
- Les labels et le masque des contours des superpixels sont obtenus.
- L'image segmentée est copiée en utilisant le masque et retournée.

3 Fonction principale main

La fonction principale gère les arguments en ligne de commande, charge l'image, appelle la fonction de segmentation et affiche les résultats.

```

1 int main(int argc, char** argv) {
2     if (argc != 3 && argc != 4) {
3         cout << " Incorrect number of arguments." << endl
4         ;
5         printHelp(string(argv[0]));
6     }
7 }

```

```

5         return EXIT_FAILURE;
6     }
7
8     const string imageFilename = string(argv[1]);
9     const string groundTruthFilename = (argc == 4) ?
10         string(argv[3]) : string();
11     const int k = stoi(argv[2]);
12
13     if (k < 1) {
14         cout << " k must be a positive integer" << endl;
15         printHelp(string(argv[0]));
16         return EXIT_FAILURE;
17     }
18
19     bool useTrust = !groundTruthFilename.empty() && k ==
20         2;
21
22     {
23         cout << endl << " Program called with the
24             following arguments:" << endl;
25         cout << " \timage file: " << imageFilename <<
26             endl;
27         cout << " \tk: " << k << endl;
28         if (useTrust)
29             cout << " \tground truth segmentation: " <<
30                 groundTruthFilename << endl << endl;
31     }
32
33     Mat image = imread(imageFilename);
34     if (image.empty()) {
35         cout << "Could not open or find the image" <<
36             endl;
37         return EXIT_FAILURE;
38     }
39
40     Mat superpixels = superpixel(image, k);
41     imshow("Superpixels", superpixels);
42     waitKey(0);
43
44     return EXIT_SUCCESS;
45 }

```

Listing 3: Définition de la fonction main

Description:

- La fonction vérifie le nombre d'arguments et affiche un message d'aide si

nécessaire.

- Elle charge l'image spécifiée par le chemin du fichier.
- Si l'image ne peut pas être chargée, elle affiche un message d'erreur et termine le programme.
- La fonction appelle `superpixel` pour segmenter l'image et affiche le résultat.

4 Choix de Conception

- Utilisation de la bibliothèque OpenCV pour la manipulation des images et la segmentation.
- Conversion de l'image en espace de couleur Lab pour une meilleure précision de la segmentation.
- Utilisation de l'algorithme SLICO pour la segmentation des superpixels.
- Paramètres configurables pour le nombre de superpixels, la compacité et le nombre d'itérations.

5 Conclusion

Ce rapport a présenté une implémentation en C++ de la segmentation d'images par superpixels en utilisant OpenCV. La conception et l'implémentation ont été expliquées en détail, y compris les fonctions utilisées, leurs paramètres et les choix de conception.