Amine AISSAOUI 22113633

TP - Semaine 4 : Opérateurs connexes et segmentation
github: https://github.com/mektoubmylove/tp4_introAnalyseImage

Exercice 1 : Etiquetage en composantes connexes

On implémente la fonction parcoursCC en suivant l'algorithme. Et puis pour ce qui est de ccLabel, on charge l'image en niveau de gris, on la binarise, on parcourt les pixels de l'image et si on trouve un pixel blanc qui n'a pas encore été traité, on utilise parcoursCC pour explorer la composante connexe de ce pixel. Une fois qu'on a identifié une composante connexe, on lui attribue une couleur aléatoire pour faciliter la visualisation des différentes composantes. Et on peut obtenir ce genre de résultat:

Exercice 2 : Filtre d'aire

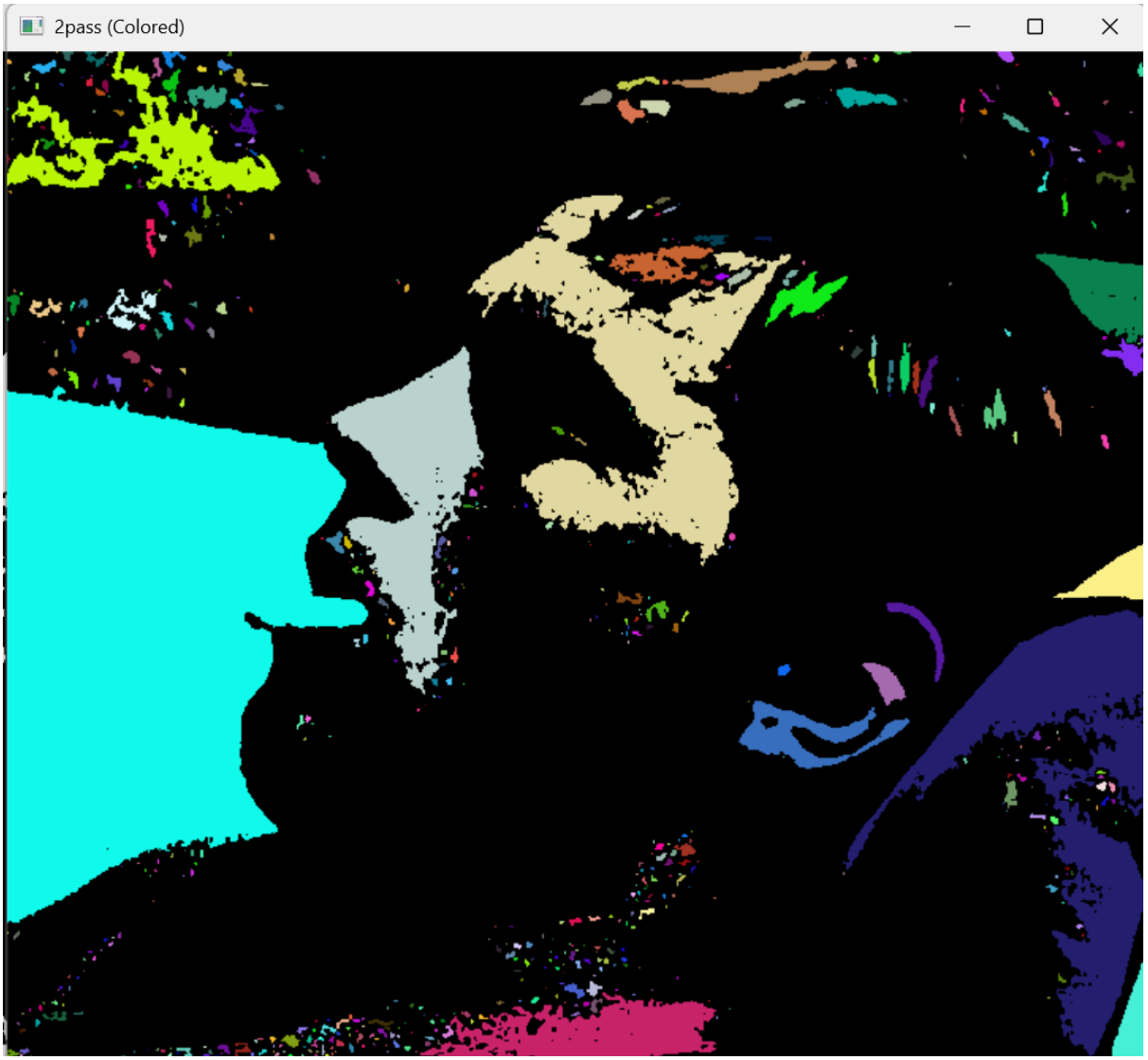
On modifie parcoursCC de manière à récupérer la taille de la composante. On suit le même raisonnement que pour la fonction précédente. Une fois toutes les composantes identifiées, on crée une nouvelle image filtrée où seules celles ayant une taille supérieure ou égale à la valeur donnée (size) sont conservées. on obtient par exemple:

pour size=100

on constate bien la suppression de petites composantes connexes

Exercice 3 : Etiquetage en composantes connexes - Mieux

Pour implémenter l'algorithme Two-Pass de labélisation des composantes connexes,on suit une approche structurée en deux étapes principales. Tout d'abord, lors de la première passe, l'image binaire est parcourue ligne par ligne. À chaque pixel de valeur 1, on vérifié ses voisins Nord et Ouest pour récupérer les labels déjà attribués.
Si aucun voisin n'était labellisé, un nouveau label était créé.
Sinon, le plus petit label des voisins était attribué au pixel actuel et on stock les relations d'équivalence entre labels .
Ensuite, lors de la deuxième passe, chaque label est remplacé par son représentant final (au lieu de remonter un à un jusqu'au représentant du label, on raccourcit directement le chemin en mettant à jour tous les nœuds intermédiaires pour pointer directement vers lui.) on peut obtenir :



Exercice 4 : Seuillage par histogramme et méthode d’Otsu

https://fr.wikipedia.org/wiki/M%C3%A9thode_d%27Otsu

On calcule l’histogramme et les probabilités de chaque niveau d’intensité

On teste tous les seuils possibles de 0 à 255. Pour chaque seuil, on divise l'image en deux groupes : Les pixels plus sombres que le seuil et ceux plus clairs.
On calcule ensuite la moyenne de chaque groupe et la variance qui mesure à quel point les pixels sont bien séparés.
Le seuil qui donne la meilleure séparation donc la plus grande variance est choisi comme seuil optimal.
On crée une image binaire, tous les pixels inférieur au seuil deviennent noirs (0) et ceux supérieur au seuil deviennent blancs (255).