

UNIVERSITÉ NATIONALE DU VIETNAM À HANOÏ
INSTITUT DE LA FRANCOPHONIE POUR
L'INNOVATION

Option : Systèmes Intelligents et Multimédia (SIM)

Promotion : 23

Année académique : 2018-2019

Module : TRAITEMENT D'IMAGES

Chaîne complète de traitement d'images

TP2

Présentés par :
SOUMANA HAMADOU ABDOURAHMANE
ESSO DISSIRAMA
PHILIPPE JEAN MITH

Prof :Dr NGUYEN Thi Oanh

Table des matières

0.1	Introduction	3
0.2	Consigne de l'application	3
0.3	PRÉ-SEGMENTATION	3
0.4	L'ÉTAPE DE SEGMENTATION	3
0.5	POST-SEGMENTATION	9
0.6	Conclusion	14

Table des figures

1	Image1 initiale	5
2	image1 grise	5
3	image1 masque	5
4	Image2 initiale	6
5	image2 grise	6
6	image2 masque	6
7	Image3 initiale	7
8	image3 grise	7
9	image3 masque	7
10	Image4 initiale	8
11	image4 grise	8
12	image4 masque	8
13	Image1 initiale	10
14	image1 après segmentation	10
15	image1 post-segmentation	10
16	Image2 originale	11
17	image2 après-segmentation	11
18	image2-post-segmentation	11
19	Image3 originale	12
20	image3 après-segmentation	12
21	image3-post-segmentation	12
22	Image4 originale	13
23	image4 après-segmentation	13
24	image4-post-segmentation	13

0.1 Introduction

La segmentation est une étape primordiale en traitement d'image. En effet, il existe de nombreuses méthodes de segmentation, que l'on peut regrouper en quatre principales classes. Ce TP nous donne l'occasion de mettre en pratique quelques techniques de segmentation en pratique. L'objectif est la présentation d'une chaîne complète de segmentation. Pour y arriver nous avons implémenté un programme en C++. Dans les lignes qui vont suivre nous présenterons les méthodes choisies, la construction des images de référence et les analyses des résultats obtenus. La constitution d'une telle chaîne complète de segmentation est la suivante :

quelques notions sur les expressions suivantes :

PRÉ-SEGMENTATION : cette phase consiste à préparer l'image à segmenter.

SEGMENTATION : une première procédure de sélection rapide des pixels d'intérêt à base de seuillage, suivie éventuellement d'une procédure de filtrage rapide également à partir d'opérateurs agissant sur image binaire. Ces deux procédures sont dites de Bas Niveau : elles admettent en entrée et en sortie une image.

POST-SEGMENTATION : Pendant cette phase nous allons corriger (régularisation le contour, remplissage des trous et la réduction des bruits) les contours des régions détectés lors de la phase précédente (segmentation) afin qu'on puisse faire une bonne analyse.

0.2 Consigne de l'application

Pour ce faire il suffit seulement de se positionner en ligne de commande dans le répertoire cible, on tape la commande `make` et enfin on exécute le programme en tapant la commande suivante : `./group5_tp2` et suivre les instructions dans le terminal (c'est-à-dire donner le nom ou le chemin complet de l'image).

0.3 PRÉ-SEGMENTATION

Dans cette partie, il est nécessaire de faire la pré-segmentation (préparation de l'image à l'entrée) avant de faire la segmentation, nous disposons déjà des images de bonne qualité, c'est-à-dire sans bruits quelconque, tandis que pour segmenter une image avec la fonction « `threshold` », l'image doit être en niveau de gris.

0.4 L'ÉTAPE DE SEGMENTATION

La segmentation regroupe toutes les techniques permettant de faire apparaître sur l'image les objets ou les entités d'intérêts. Ces techniques sont la détection de contour et le seuillage. Dans une image un contour est un ensemble de

points qui correspond à un changement rapide (vis à vis du voisinage des points du contour) du niveau. Nous utiliserons en deux étapes une approche classique pour ce traitement : — Une accentuation des contours ;

— Un seuillage qui permet d'obtenir une image binaire.

Pour déterminer les différents contours qui composent une image, nous avons utilisé l'algorithme d'OTSU pour notre étude. Les images que nous disposons sont en couleur et seront converties en niveau de gris afin de les passer à la fonction « threshold » de opencv qui permet d'appliquer l'algorithme d'OTSU pour faire un seuillage des contours en fin d'avoir une image binaire. Les images en niveau de gris et les masques obtenus après la segmentation de ces différentes images sont illustrés ci-dessous :

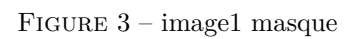
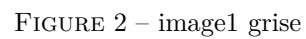
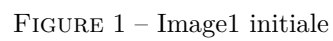




FIGURE 4 – Image2 initiale

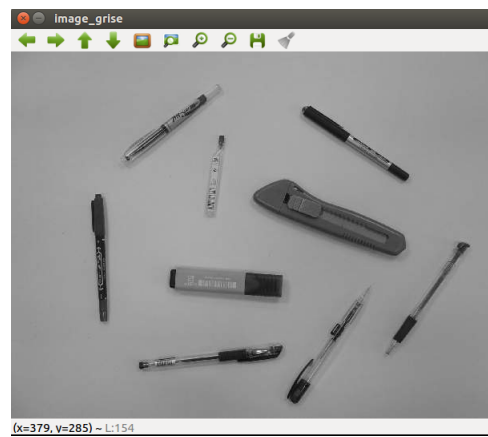


FIGURE 5 – image2 grise

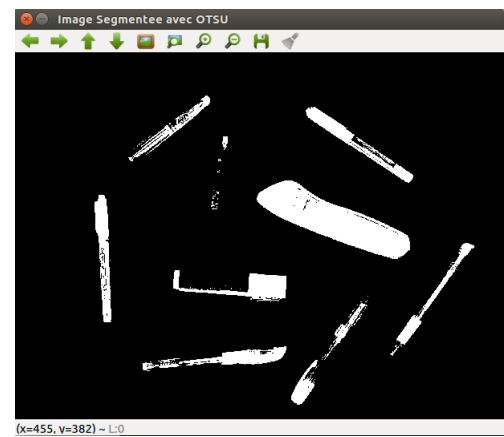


FIGURE 6 – image2 masque



FIGURE 7 – Image3 initiale



FIGURE 8 – image3 grise

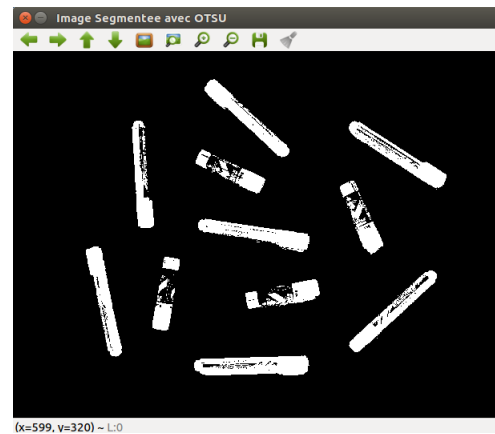


FIGURE 9 – image3 masque



FIGURE 10 – Image4 initiale



FIGURE 11 – image4 grise

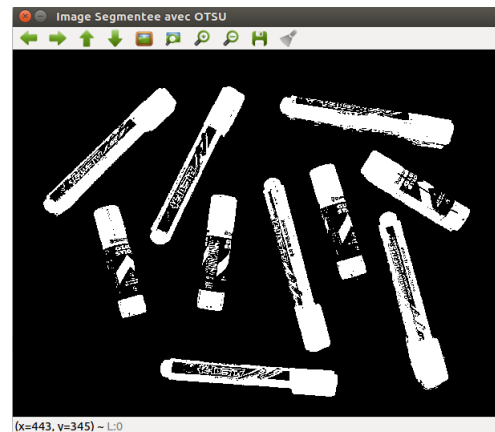


FIGURE 12 – image4 masque

Ici nous avons deux types d'images de chaque objets à savoir :l'image en niveau de gris et l'image masque de segmentation.En ce qui concerne les images en niveau de gris on constate que les contours sont bien formés sauf pour les images de couleurs transparentes ont leurs contours légèrement réduit. Donc nous pouvons dire que cette fonction est moyennement acceptable pour la formation des contours et permettra de nous fournir des images sollicitées pour notre étude. Notre seconde image est une image binaire. La binarisation est une opération de traitement simple qui consiste à ne conserver dans une image que deux(2) niveaux de gris (0=noir 1=blanc) ;

— soit en convertissant en noir tous les niveaux de gris inférieurs à un seuil et en blanc tous ceux qui dépassent le seuil.

— soit en convertissant en blanc toutes les valeurs comprises entre deux seuils, et en noir tout le reste.

Le choix de seuil est une étape très nécessaire pour avoir une bonne segmentation. A cet effet, nous avons effectué des différents tests sur plusieurs seuils en appliquant sur nos images en fin de choisir ce seuil meilleur. De ce fait, malgré le seuil choisi sur les variants valeurs de test, on constate que, dans certains cas, notre choix du seuil est loin d'être optimal. En particulier, lorsque la surface des objets est petite(plusieurs objets dans la surface) vis-à-vis de celle du fond et les objets ont de petit taille, certains contours sont fortement décalés du côté du "pic" le plus important (objet 1). Par ailleurs si l'objet contient des images de grande taille et moins de couleurs, on constate que le résultat est moyen (contour bien formé et la distinction entre le contour et le fond est très claire) (cas objet 4).

0.5 POST-SEGMENTATION

Il est vrai que la segmentation n'est pas toujours parfaite c'est-à-dire qu'il y aura toujours quelques failles d'où la nécessité de la phase de post-segmentation pour corriger les failles. A cet effet nous présenterons dans cette partie les techniques pour corriger les images segmentées avec des erreurs. pour ce faire il faut appliquer les opérations morphologiques de base : la dilatation et l'érosion, cette composition est appelée la fermeture ou l'ouverture selon l'ordre d'application de ces opérations.Ces deux techniques nous permettent de procéder à la correction des masques dans le but de lisser nos images.

Ainsi toutes les régions détectées seront étiquetées, avec des couleurs différentes et nous permettra d'analyser les images d'origine et les masques dans le but de montrer uniquement les régions détectées.

En pratique nous avons utilisé les fonctions d'opencv « dilate » et « erode » nous permettant d'appliquer ces techniques (la dilatation et l'érosion) et la fonction « drawcontours » pour d'attribuer l'étiquetage des couleurs.

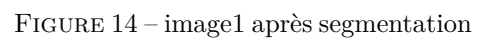
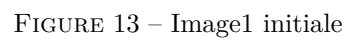




FIGURE 16 – Image2 originale

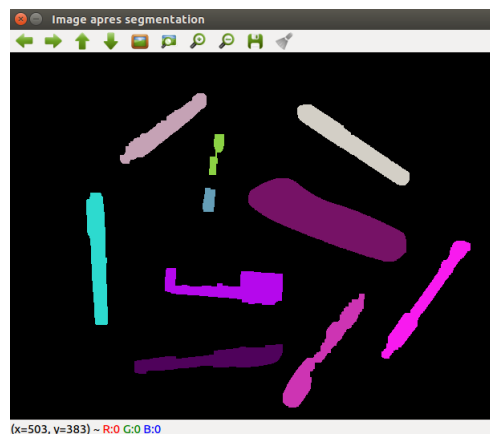


FIGURE 17 – image2 après-segmentation

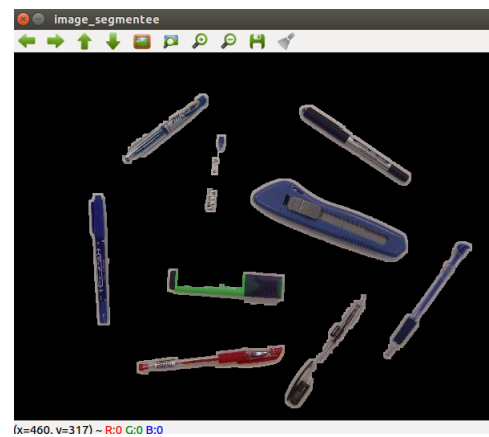


FIGURE 18 – image2-post-segmentation



FIGURE 19 – Image3 originale

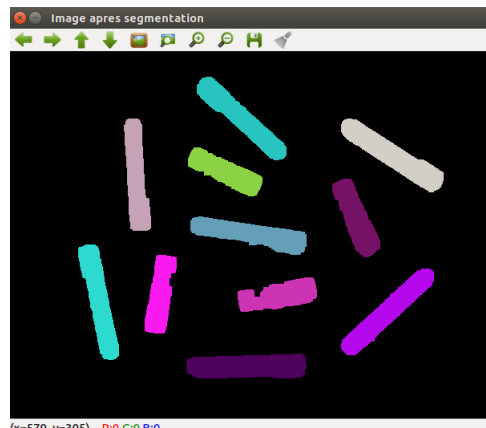


FIGURE 20 – image3 après-segmentation



FIGURE 21 – image3-post-segmentation



FIGURE 22 – Image4 originale

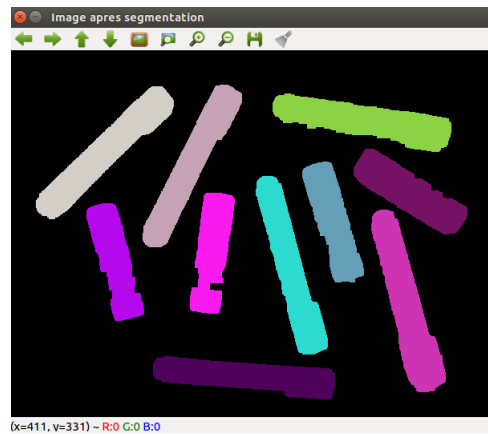


FIGURE 23 – image4 après-segmentation



FIGURE 24 – image4-post-segmentation

Observations/Analyses :

Nous constatons que les trous sont comblés et les contours des régions lissés (figures 20 et 21). L'étiquetage est bien réalisé car toute région dans le résultat de la segmentation est marquée par une couleur bien déterminée. Mais d'autres régions restent mal segmentées (figures 16 à 21).

Les contours des régions segmentées sont moins précis vu qu'ils intègrent plus le fond des images, Ce résultat est le fruit de l'opération de fermeture appliquée aux images qui proviennent de la segmentation. En effet la fermeture est une opération qui commence par la dilatation et se termine par une érosion. La dilatation permet de fermer les trous afin de rendre les contours lisse mais tend aussi à fausser le tracé des contours des régions et donc à intégrer des pixels appartenant au fond d'image c'est la raison pour laquelle on applique également une érosion dans le but de corriger cette insuffisance de la dilatation.

Ainsi, en analysant ces différentes images, l'objets3 et l'objets4 leur post-segmentation est presque parfait, nous pouvons dire que plus l'image contient moins de couleur le post-segmentation se fait correctement cependant l'objets1 et l'objets2 ont du mal à traiter puisque qu'ils contiennent plus couleurs différentes.

0.6 Conclusion

En définitive, ce travail nous a conduit de concevoir et de réaliser une chaîne de traitement d'images en utilisant les techniques citées ci dessus, à savoir le **pré-segmentation**, la **segmentation** et la **post-segmentation**. Au niveau de pré-segmentation, nous n'avons pas fait grande chose vu la qualité des images offerts mais nous avons transformé les images au niveau de gris puis au binaire pour pouvoir appliquer certaines fonctions openCv qui l'exige au niveau segmentation. Au niveau de la segmentation nous avons appliqué l'algorithme de OTSU pour faire la segmentation. Et enfin au niveau de post-segmentation nous avons procédé l'application fermeture pour combler certains trous et bien formé certains contours. Après l'études de ces différentes techniques, les résultat nous montre que l'algorithme OTSU a du mal à traiter certaines images telles que : les images à couleurs transparent ou les images qui comportes plusieurs couleurs. par ailleurs cet algorithme est très efficace si l'image comporte une forte contraste. Ainsi on peut dire que comme il n'existe pas un algorithme parfait sur tous les plans, nous pouvons dire que OTSU est efficace pour la segmentation des images compte tenu des résultats qu'il fourni.