

Traitement d'images

RAPPORT

IFI-Promotion :23

THÈME :CHAÎNE COMPLÈTE DE TRAITEMENT D'IMAGES

Rédiger par :

M. LAMAH RICHARD

M. CISSOKO MAMADOU

BEN HAMIDOU

Encadrant :

Mme NGUYEN THI

OANH

Table des matières

I	Introduction :	2
II	Description de la chaîne de traitement :	2
III	Fonctionnement du programme :	3
III.1	Expérimentations et résultats :	3
III.1.1	Phase de segmentation	3
III.1.2	POST-SEGMENTATION :	6
IV	Discussion :	10
V	Conclusion :	11

I Introduction :

Le but de ce TP3 est de définir une chaîne complète de traitement d'images centrée sur la segmentation et d'appliquer les traitements de ladite chaîne à une série d'images puis d'analyser les résultats obtenus. **Consigne du TP3 :** Description claire de la chaîne de traitement complète que vous avez définie, en expliquant l'ordre des traitements, les paramètres utilisés, et toutes les explications sur vos choix. Donnez aussi vos résultats sur toute la série des images d'objets en analysant l'efficacité de vos traitements. Votre analyse compte autant sinon plus que vos résultats seuls. Le programme que vous remettez n'a pas d'arguments, sauf l'image à traiter. C'est le même programme (sans aucun paramètre autre que le nom de l'image) pour les 4 images et il doit donner le même résultat pour toutes les images.

II Description de la chaîne de traitement :

La chaîne de traitement que nous avons défini est structurée en deux phases :

- la phase de segmentation où les différentes régions correspondantes aux objets contenus dans l'image sont identifiées et représentées par des masques ;
- la phase de post-segmentation durant laquelle des améliorations sont apportées à la forme des régions détectées lors de la phase de segmentation (correction des contours, remplissage des trous) et le bruit réduit au maximum.

Notons qu'à ces deux phases nous aurions pu ajouter une phase de pré-segmentation qui aurait consisté à préparer les images à la segmentation toutefois nous ne l'avons pas fait car les images qui nous ont été fournies sont de bonne qualité sans bruit apparent, avec un éclairage correct et un fond relativement uni.

III Fonctionnement du programme :

Afin d'appliquer les traitements précédemment décrits, nous avons implémenté un programme nommé « **segmentation.cpp** » qui prend en paramètre le nom de l'image à traiter. Pour pouvoir utiliser les fonctionnalités de ce programme il est nécessaire de suivre les étapes suivantes :

1. Se placer dans le dossier contenant le code source du programme
2. Copier les images à segmenter directement dans le dossier contenant le code source.
3. Lancer le terminal
4. Exécuter la commande « **make** » pour compiler le programme
5. Taper la commande : **./segmentation**
6. Taper le nom de l'image **nom_iimage.extension**(*Exemple : objets1.jpg*)

III.1 Expérimentations et résultats :

III.1.1 Phase de segmentation

Pour segmenter les régions contenues dans les images, nous avons utilisé l'algorithme de OTSU qui permet de réaliser une segmentation en région avec seuillage. Notons que le seuil optimal est calculé de façon automatique en minimisant la variance intraclasse. Nous avons choisi de garder l'espace de couleur RVB (Rouge Vert Bleu) mais l'image a due être convertie en niveau de gris afin d'être passée à la fonction « **threshold** » de opencv permettant d'appliquer l'algorithme de OTSU. Les masques obtenus après segmentation des différentes images sont présentés ci-dessous et sont en blanc (rappelons que ces masques ne sont pas les résultats définitifs de la segmentation, nous présenterons plus bas les masques définitifs en couleurs)

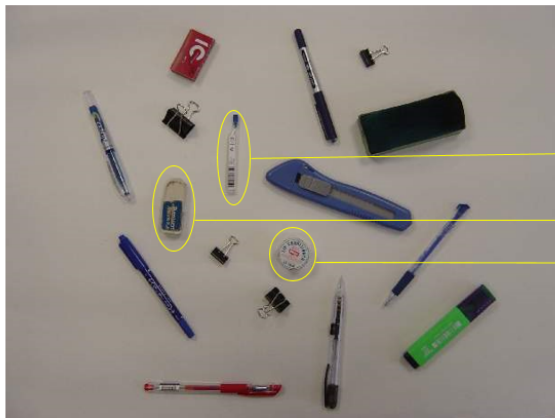


Figure 1 : Image originale Objets1.jpg

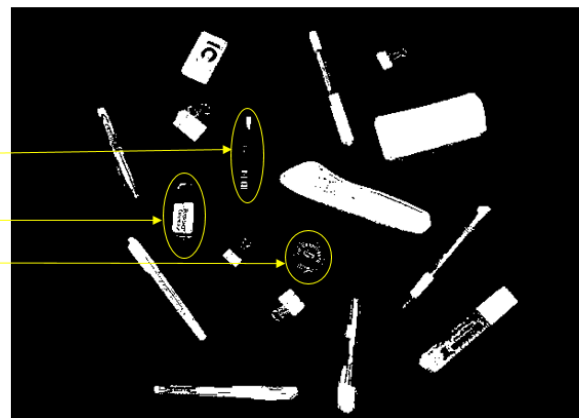


Figure 2 Masque de Segmentation de l'image Objets1.jpg

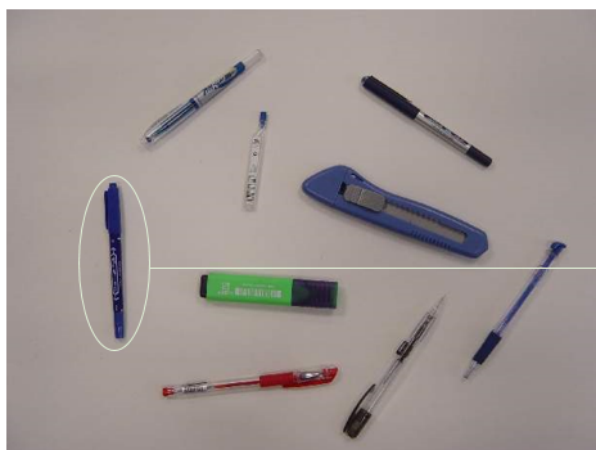


Figure3 : Image originale Objets2.jpg



Figure4 : Masque de Segmentation de l'image Objets2.jpg

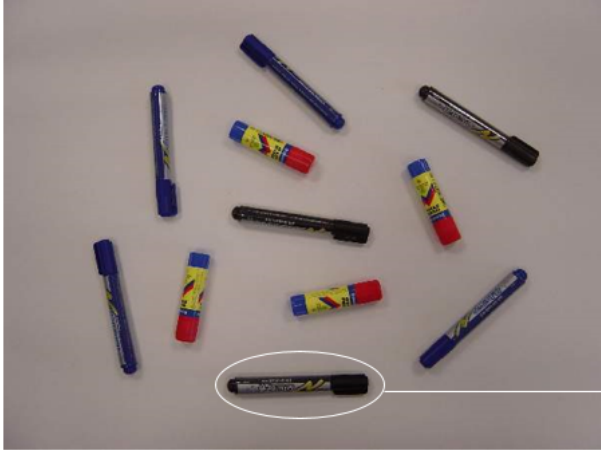


Figure5 : Image originale Objets3.jpg

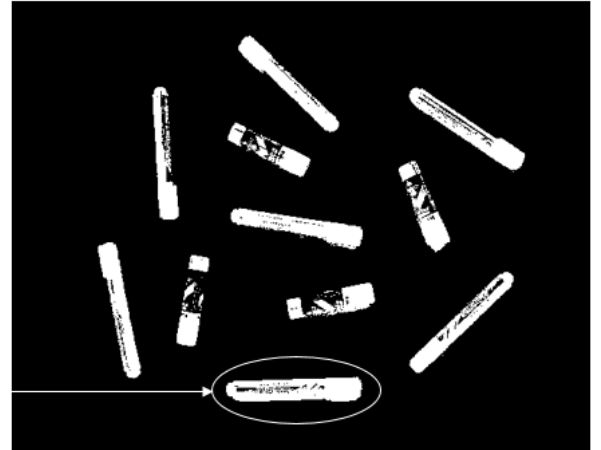


Figure6 : Masque de Segmentation de l'image Objets3.jpg



Figure7 : Image originale Objets4.jpg



Figure8 : Masque de Segmentation de l'image Objets4.jpg

L'analyse des images des masques montre que l'algorithme de OTSU permet de réaliser une assez bonne segmentation des régions et donc des objets. En effet tous les objets présents dans chacune des images ont été détectés et représentés dans les masques toutefois cette détection n'est pas optimale dans tous les cas vu que l'on note la présence de trous dans les masques de certaines régions mais

aussi des contours irréguliers voir incomplets. Plus spécifiquement les objets ayant des couleurs unies et différentes du fond de l'image sont bien segmentés (zones encerclées en blanc figure 3 à 6) tandis que les objets qui ont des couleurs proches de celle du fond ou qui sont transparents sont assez mal segmentés (zones encerclées en jaune figure 1 et 2). On note aussi que les ombres des objets sont intégrées à des régions. Ces observations s'expliquent par le principe de l'algorithme de OTSU dont l'objectif est de diviser l'histogramme des images en deux sections séparées par une valeur de seuil. Les valeurs des pixels d'un côté du seuil sont considérées comme le fond et les valeurs de l'autre côté du seuil sont considérées comme des régions. Ainsi, si des objets ont des parties dont des couleurs sont proches de celle du fond de l'image alors ces parties seront aussi considérées comme le fond de l'image lors de la segmentation puisqu'elles seront du côté du seuil correspondant au fond. A l'opposé les objets qui ont des couleurs suffisamment différentes du fond seront facilement considérées comme des régions puisque les valeurs des pixels correspondants à ces parties seront de l'autre côté du seuil.

III.1.2 POST-SEGMENTATION :

La segmentation n'étant pas parfaite, dans cette partie du TP, l'application des techniques de dilatation et d'érosion nous permettront de procéder à la correction des masques dans le but de lisser nos objets. L'ordre d'application de ces techniques sera le même que celui cité plus. Dilatation ensuite érosion ce qui nous permet d'effectuer une fermeture. Toutefois les régions détectées seront étiquetées, des couleurs différentes leurs seront affectées et nous analyserons les images d'origine et les masques dans le but de montrer uniquement les régions détectées. Les principales fonctions d'opencv nous permettant d'appliquer ces techniques sont Entre autres « **dilate** » pour la dilatation, « **erode** » pour l'érosion, « **draw-contours** » pour l'étiquetage et l'attribution des couleurs.



Figure9 : Image Objets1.jpg post-segmentation



Figure10 : Éléments détectés pour l'image objets1.jpg

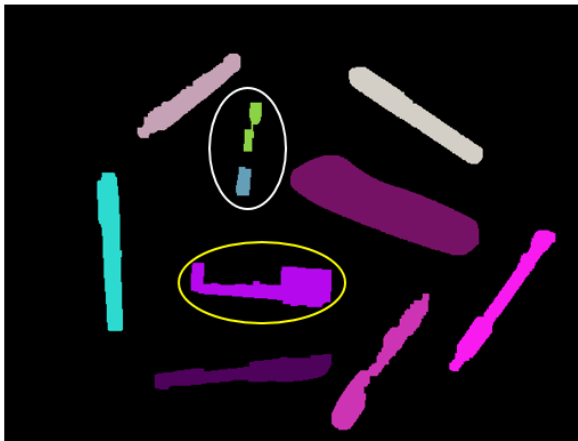


Figure11 : Image Objets2.jpg post-segmentation

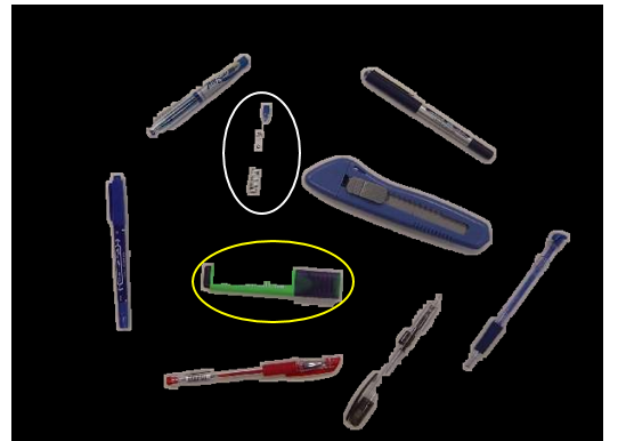


Figure12 : Éléments détectés pour l'image objets2.jpg



Figure13 : Image Objets3.jpg post-segmentation

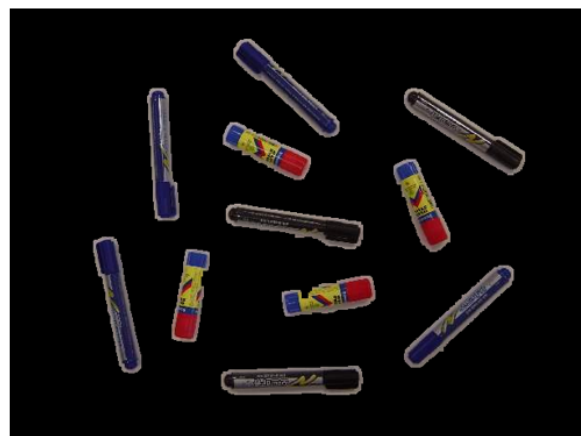


Figure14 : Éléments détectés pour l'image Objets3.jpg



Figure15 : Image Objets4.jpg post-segmentation



Figure16 : Éléments détectés pour l'image Objets4.jpg

Compte tenue des résultats affichés sur les images, nous pouvons affirmer que les résultats obtenus montrent que les techniques de correction appliquées sont efficaces ceci par le simple fait que les trous sont comblés et les contours des régions lissés. L'étiquetage est bien réalisée car toute région dans le résultat de la segmentation est marquée par une couleur bien déterminée. Mais d'autres régions restent mal segmentées (zones encerclées en blanc et bleu figures 9 à 12). 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 tends 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. Cependant lorsque le type et/ou la taille des éléments structurants choisis pour effectuer la dilatation et l'érosion ne sont pas les mêmes l'effet d'une des opérations peut être dominant sur celui de l'autre. C'est notre cas ici car nous avons utilisé un noyau de taille 13×13 pour la dilatation et de taille 7×7 pour l'érosion. Ce choix se justifie par le fait que lorsqu'on utilise la même taille de noyau pour les deux opérations, les contours sont certes plus proches des objets mais l'érosion fait passer une bonne partie des régions en images de fond surtout pour les images objets1.jpg et objets2.jpg. Les figures ci-après montrent les résultats de poste segmentation pour une dilatation et une érosion utilisant le même élément structurant de taille 13×13 pour les images objets1.jpg et objets2.jpg ; les résultats pour les images objets3.jpg et objets4.jpg restent pratiquement les mêmes.



Figure17 : *Image objets1.jpg post-segmentation*



Figure18 : *Éléments détectés pour l'image objets1.jpg*

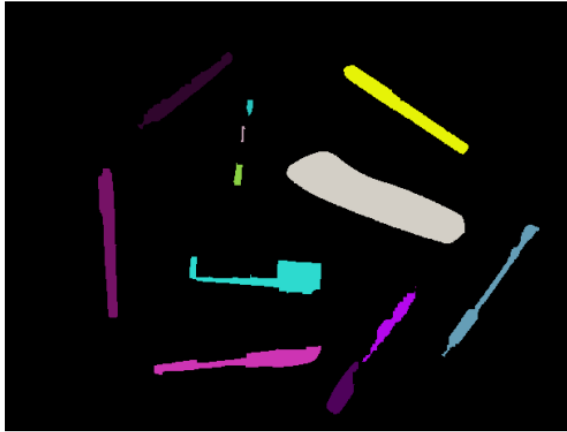


Figure19 : *Image objets1.jpg post-segmentation*

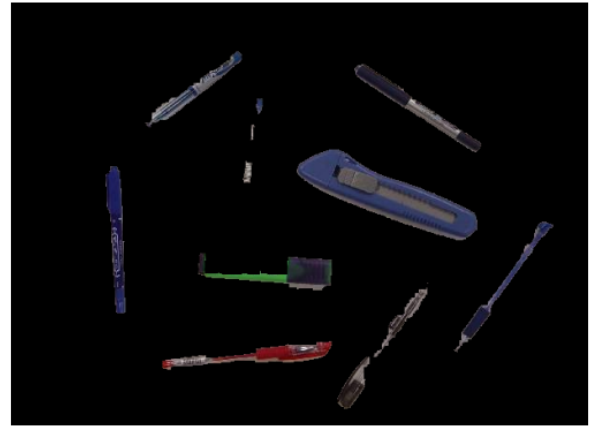


Figure20 : *Éléments détectés pour l'image objets1.jpg*

IV Discussion :

Nous retenons des expérimentations menées et de leurs résultats montrent que notre chaîne de traitement est efficace puisqu'elle permet d'identifier tous les objets régions présents dans les différentes images. Cependant elle n'est pas optimale

parce que certains objets principalement ceux transparents et qui portent des couleurs proches du fond de l'image même si ils sont détectés, sont mal segmentés. Nous pouvons noter aussi que la chaîne fonctionne bien pour les images 3 et 4 car elles ne comportent pas d'objets transparents tandis que pour les images 1 et 2 les résultats sont moins bons.

V Conclusion :

En résumé ce TP nous aura permis de concevoir une chaîne de traitement d'images comportant deux phases : la segmentation et la post-segmentation, la pré-segmentation ayant été ignorée en raison de la qualité des images fournies. Pour la phase de segmentation nous avons employé l'algorithme de OTSU puis nous avons appliqué une fermeture en post-segmentation. Nous avons observé que l'algorithme de OTSU ne permet pas de bien segmenter les objets transparents ou dont la couleur est proche de celle du fond. Toutefois il est globalement assez efficace pour la segmentation des images ou il existe un très fort contraste du point de vue de l'intensité entre le fond et les objets à segmenter. La fermeture appliquée en post-segmentation aura quant-à elle permis de combler les trous et de régulariser les contours (Lisser les contours) mais au pris d'un moins bon tracé du contour des objets. Notons qu'en raison de la qualité des images fournies, la pré-segmentation a été ignorée. Ceci étant dit nous aurions pu tout de même ajuster le contraste de façon à avoir une différence d'intensité plus grande entre le fond et les objets dans les images. Mais un tel traitement n'aurait pas change énormément le résultat au niveau de la segmentation.