

TP 3 : CHAÎNE COMPLÈTE DE TRAITEMENT D'IMAGES

Réalisé par :

- MEDOU Daniel Magloire, IFI-Promotion 21
- MILORME Pierre Rubens, IFI-Promotion 21
- MALLE Zoumana, IFI-Promotion 21

But du TP : Dans ce dernier, il est question pour nous de définir une chaîne complète de traitement d'images centrée sur la segmentation tout en appliquant ces traitements chaîne à une série d'images et enfin analyser le rendu de l'image originale.

Consigne du TP : 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.

◆ Description de la chaîne de traitement

La structuration de la chaîne de traitement ici se fera à deux niveau :

- ✓ **le niveau de segmentation** : ici, toutes les régions correspondantes aux objets contenus dans l'image sont bien identifiées et représentées par des masques;
- ✓ **le niveau de post-segmentation** : Ici le traitement nous donne des améliorations au niveau de la forme des régions détectées au cours du traitement appliquée à la segmentation (correction des contours, remplissage des trous) et le bruit considérablement réduit.

Il est essentiel pour nous de signaler que la pré-segmentation qui devrait être faite bien avant pour la préparation des images à la segmentation proprement dite ne se fera plus car le set d'image obtenu est d'une qualité parfaite (fond unique et un bon contraste) car nous n'observons pas de bruits apparents.

◆ Fonctionnement du programme

Pour utiliser le programme implémenté, l'utilisateur devrait suivre les étapes suivantes :

- ✓ se placer dans le dossier contenant le code source du programme ;
- ✓ copier les images à segmenter directement dans le dossier contenant le code source.
- ✓ lancer le terminal ;
- ✓ exécuter la commande « *make* » pour compiler le programme ;
- ✓ taper la commande : *./tp3*
- ✓ Taper le nom de l'image *nom_image.extension* (Exemple : *objets1.jpg*)

TESTS ET RÉSULTATS

→ SEGMENTATION

La segmentation des régions contenues dans les images sera faite grâce à l'algorithme de OTSU. Ce dernier nous permettra d'effectuer une segmentation en région avec seuillage. Dans notre cas, le seuil optimal sera automatiquement calculé de façon à minimiser la variance intra-classe. L'espace de couleur RVB (Rouge Vert Bleu) est conservé cependant, l'image subira une conversion en niveau de gris pour être accessible à la fonction « *threshold* » de opencv applicable par l'algorithme de OTSU. Les masques obtenus après l'opération de segmentation des images sont présentés ci-dessous et sont en blanc. Il est important pour nous de rappeler que ces masques ne sont pas les rendus définitifs de la segmentation, ils seront présentés plus bas ces masques définitifs et en couleurs):

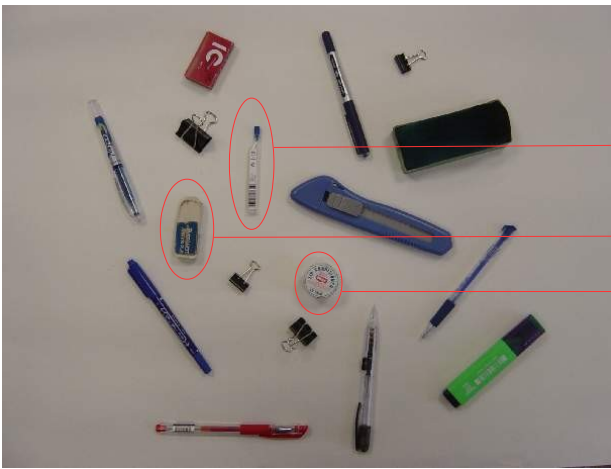


Figure1 : Image originale objets1.jpg

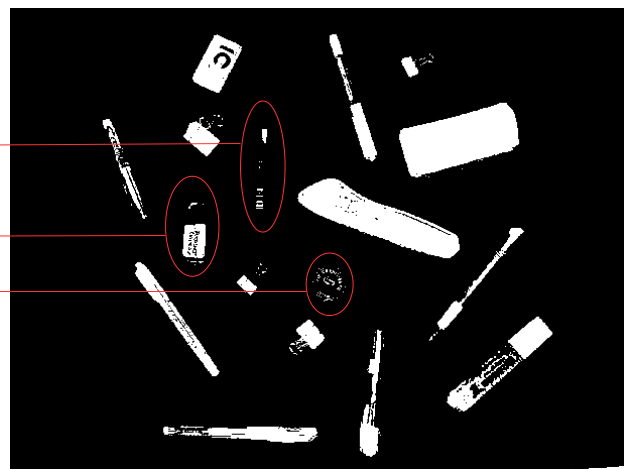


Figure2 : Masque de segmentation de l'image objets1.jpg



Figure3 : Image originale objets2.jpg



Figure4 : Masque de segmentation de l'image objets2.jpg



Figure5 : Image originale objets3.jpg

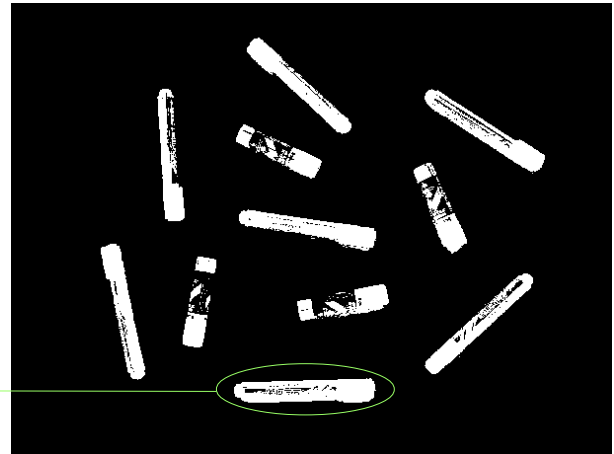


Figure6 : Masque de segmentation de l'image objets3.jpg



Figure7 : Image originale objets4.jpg

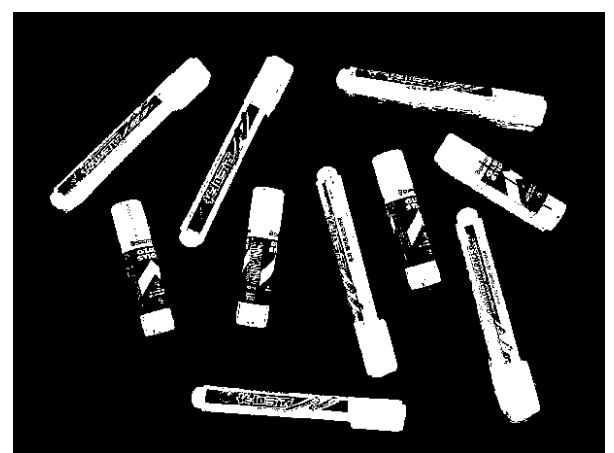


Figure8 : Masque de segmentation de l'image objets4.jpg

L'analyse des images des masques montre que l'algorithme de OTSU sur l'analyse des images de masques nous a permis de réaliser une meilleur segmentation des régions des objets. Car ces derniers présents dans chacune des images étaient détectés et représentés dans les masques et cette détection n'est pas optimale vu que nous voyons toujours des trous dans les masques de certaines régions et même des contours incomplets. De manière spécifique, ces objets ont des couleurs unies et différentes du fond de l'image sont bien segmentés (zones encadrées en vert figure 3 à 6) alors que les objets qui ont des couleurs proches de la couleur de fond ou qui sont transparents sont mal segmentés (zones encadrées en rouge figure1 et 2). Notons que les ombres des objets sont intégrées aux régions. Observations expliquées selon le principe de l'algorithme de OTSU qui a pour objectif de diviser l'histogramme des images en deux sections qui séparées par une valeur de seuil. Les valeurs des pixels d'un coté du seuil sont considérées comme le fond et les valeurs de l'autre coté 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'elle seront du coté 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.

→ 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 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 autre « *dilate* » pour la dilatation, « *erode* » pour l'érosion, « *drawcontours* » pour l'étiquetage et l'attribution des couleurs.

Expérimentations :



Figure9 : Image objets1.jpg post-segmentation



Figure10 : Éléments détectés pour l'image objets1.jpg

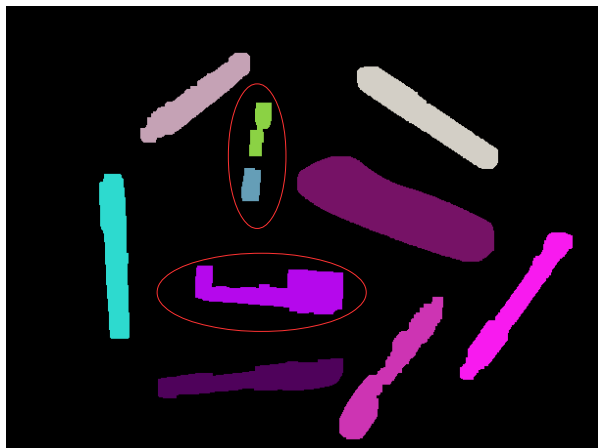


Figure11 : Image objets2.jpg post-segmentation

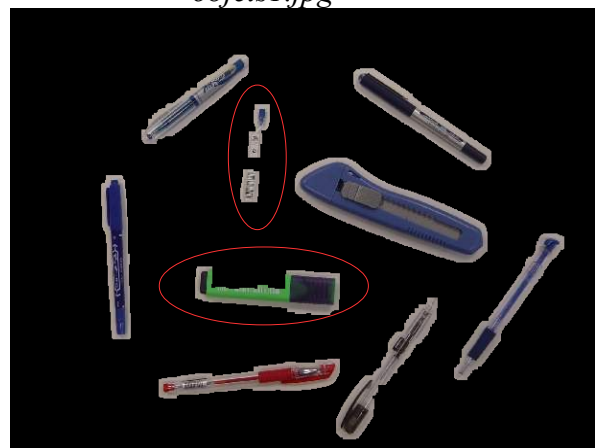


Figure12 : Éléments détectés pour l'image objets2.jpg



Figure13 : Image objets3.jpg post-segmentation

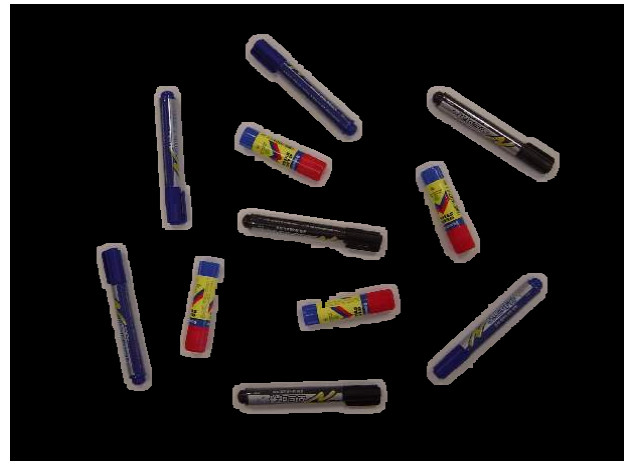


Figure14 : Éléments détectés pour l'image objets3.jpg



Figure15 : Image objets4.jpg post-segmentation



Figure16 : Éléments détectés pour l'image objets4.jpg

Compte tenu du rendu, nous pouvons certifier que les résultats dont nous obtenons montrent que les techniques de correction appliquées sont efficaces ceci par le simple fait que les trous sont comblés et les contours des régions sont lisse. L'étiquetage bien réalisé car toute région résultat de la segmentation est marquée par une couleur bien déterminée. Mais d'autres régions restent mal segmentées (zones encadrées en rouge figures 9 à 12) car celles-ci ont été mal segmentées. Les contours des régions segmentées sont moins précis vu qu'ils intègrent plus le fond d'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 permettant 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. 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 restant pratiquement les mêmes.



Figure17 : Image objets1.jpg post-segmentation



Figure18 : Éléments détectés pour l'image objets1.jpg

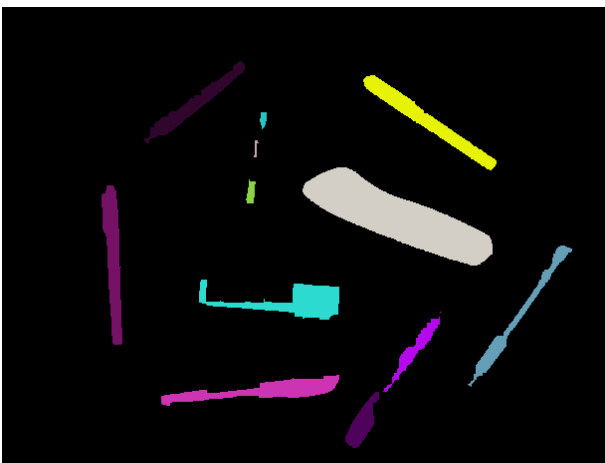


Figure19 : Image objets1.jpg post-segmentation

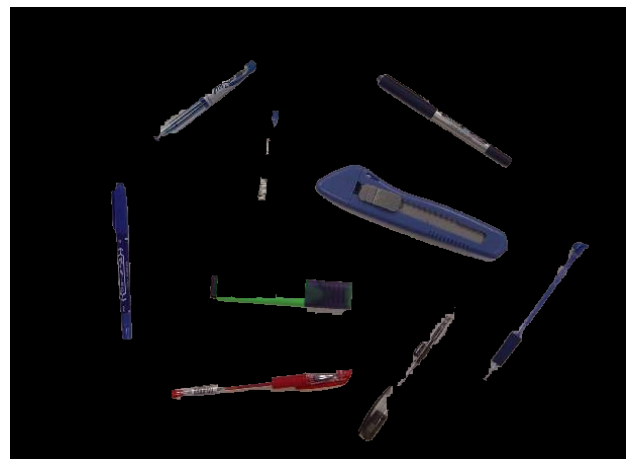


Figure20 : Éléments détectés pour l'image objets1.jpg

◆ Discussion

Nous pouvons dire que nos tests effectués et de les résultats obtenus montrent l'efficacité de notre chaîne de traitement étant donné qu'elle nous permet d'identifier tous les objets et régions présents dans les différentes images. Mais elle n'est pas optimale par le simple fait que certains objets transparents ayant des couleurs proches ou exactement à celle du fond quand bien même sont détectés aussi sont mal segmentés. Nous pouvons dire que le fonctionnement de la chaîne est bien par le rendu des images 3 et 4 car n'ayant pas d'objets transparents tandis que pour les image 1 et 2 les résultats ne sont moins satisfaisants.

Conclusion

En dernière analyse, ce TP nous a permis de concevoir et de réaliser une chaîne de traitement d'images à deux niveaux à savoir la **segmentation** et la **post-segmentation**. Pour la segmentation l'algorithme de OTSU nous a permis de la réaliser et nous avons procédé juste à l'application de la fermeture pour la post-segmentation. Les résultats nous ont montré que l'algorithme de OTSU ne permet pas d'effectuer une segmentation parfaite des objets transparents et même ceux des objets ayant comme couleur celle semblable à la couleur du fond. Tandis qu'elle est parfaite voir même efficace pour la segmentation des images. La fermeture appliquée en post-segmentation nous a permis de combler les trous et de lisser les contours. Notons qu'en raison de la qualité des images fournies, la pré-segmentation a été ignorée.