

**Institut de la Francophonie pour l'Innovation**  
**Traitement d'images**  
**TP 3 : Chaîne complète de traitement d'images**

**Préparé par groupe 12:**

ALPHONSE Rooldy, EDOUARD Grégory, GOINT Mongetro

=====

Le TP consiste à définir une chaîne complète de traitement d'images basée sur la segmentation et d'appliquer les traitements de cette chaîne sur des images, puis d'analyser les résultats obtenus.

**PARTIE I : Description de la chaîne de traitement complète définie**

Notre chaîne de traitement complète est segmentée en deux phases :

D'abord, Une phase de **segmentation** où les régions relative aux objets contenus dans l'image sont identifiées et représentées par des masques.

Ensuite, une phase de **post-segmentation** consistant à apporter des améliorations à la forme des régions détectées lors de la première phase de segmentation à travers laquelle on a eu (correction des contours, remplissage des trous) et la réduction au maximum des bruits.

**- Description et fonctionnement du programme**

Pour appliquer les traitement complète, il suffit d'abord de lancer un terminal sur Linux et de se positionner dans le répertoire contenant notre fichier « **segmentation.cpp** » qui contient les codes de notre programme; à noter que l'image à traiter doit se trouver dans ce même répertoire;

Ensuite, exécuter la commande « make » pour compiler le programme ;

Enfin, lancer la commande : « **./segmentation nom\_image** » pour appliquer la segmentation sur l'image désirée.

**Exemple: ./segmentation objet1.jpg**

**PARTIE II : Expérimentation et résultats de la chaîne de traitement**

Dans cete partie, nous tenons à présenter cas par cas les résultats (notamment les masques) obtenus après l'application d'un traitement similaire sur les quatre images (objet1.jpg, objet2.jpg, objet3.jpg et objet4.jpg) avec notre chaîne de traitement.

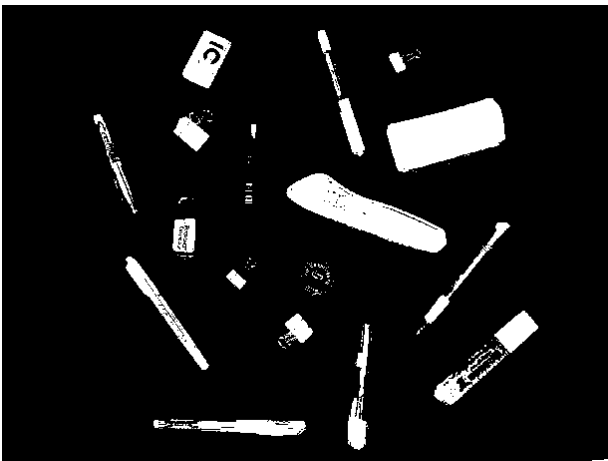
D'abord, pour l'image «*objet1.jpg*»:



**Figure1:** image d'entrée objet1



**Figure2:** Image pré-segmentée objet1



**Figure3:** Image post-segmentée objet1



**Figure4:** Image segmentée objet1

L'analyse des résultats nous montre qu'une bonne segmentation des régions et aussi des objets a été réalisée sur notre image.

D'abord, dans la **Figure2** qui représente le résultat de la pré-segmentation, nous pouvons constater que tous les objets présents dans l'image d'entrée ont été détectés et apparus, mais de manière imparfaits pour quelques un, avec de brusques changements sous forme de fusion, ce qui cause une certaine déformation de ceux-là . Et dans la **Figure3** qui représente le résultat de la post-segmentation, tous les images sont converties en images en niveau de gris. Mais l'un des constats assez important est que les objets qui ont des couleurs proches de celle du fond ont une segmentation plus imparfaite que les autres, ce qui fait que l'image segmentée dans la Figure4 paraît un peu différente avec des valeurs de pixel modifiées par rapport à l'image d'entrée.

A collection of various stationery items including pens, pencils, a marker, a stapler, and a glue stick. The items are arranged on a plain white background. There is a blue ballpoint pen, a silver ballpoint pen, a black marker, a blue stapler, a green glue stick, a red ballpoint pen, a silver ballpoint pen, and a blue pencil.

A collection of various pens and pencils, including blue, green, and red ones, arranged on a black background. The items include several ballpoint pens in blue and silver, a green pen, a red pen, and a blue pencil. There are also some small, detached pieces of the pens and pencils scattered around.

Pour les résultats l'image "objet2", nous pouvons remarquer qu'il n'y a pas trop grande différence par rapport aux résultats de l'image "objets1" sinon un peu plus de décalage au niveau des contours des différents objets.

3



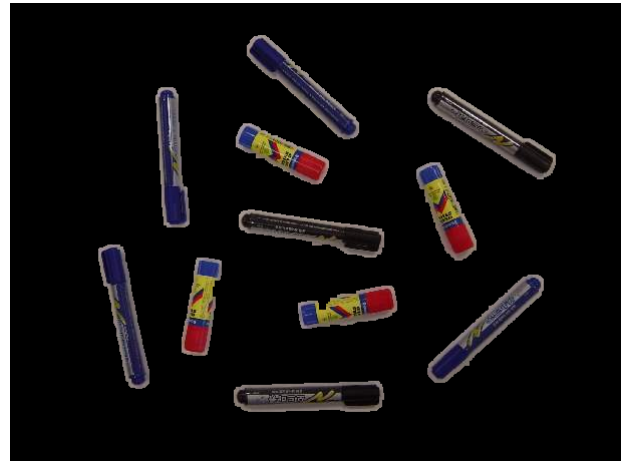
**Figure9:** image d'entrée objet3



**Figure10:** Image pré-segmentée objet3



**Figure11:**Image post-segmentée objet3



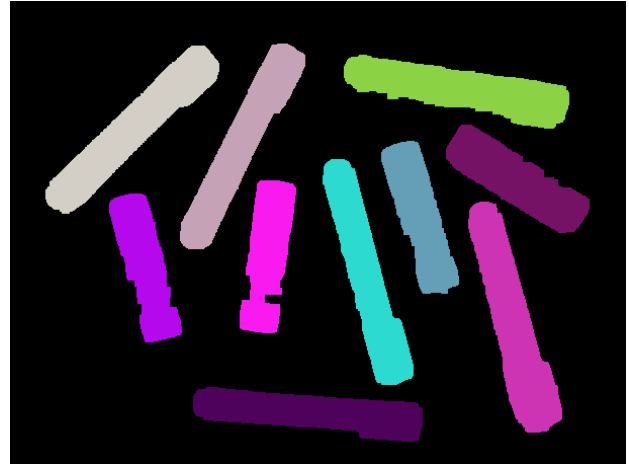
**Figure12:** Image segmentée objet3

Comparativement aux deux premiers cas, nous pouvons constater depuis l'étape de pré-segmentation, il y a des améliorations au niveau de la représentation des objets. Donc, leur contour paraissent un peu plus régulier et la forme casi intégrale des objets, tant au niveau de l'étape de prés-segmentation qu'au niveau post-segmentation est gardée. Ce qui fait que l'image segmentée paraît un peu similaire par rapport à celle qu'on avait entrée.

*Enfin, pour l'image **objet4.jpg**:*



**Figure13:** image d'entrée objet4



**Figure14:** Image pré-segmentée objet4



**Figure15:**Image post-segmentée objet4



**Figure16:**Image segmentée objet4

Pour ce dernier cas avec l'image(**objet4**), il est bon de noter que les objets sur l'image d'entrée était plus nette par rapport aux précédents cas. Donc, les résultats obtenus montrent que les corrections apportées ont été plus efficaces car les trous ont été comblés et les contours des régions sont devenus plus réguliers, même s'il y a des régions apparues avec des contours un peu ombrés. Au niveau de l'image post-segmentée(**Figure15**), nous pouvons remarquer les objets gardent leur forme initiale, et la valeur des pixels ont été conservées. Ce qui explique que les images sont un peu plus nettes comparativement aux précédents cas, et même les écrits qui étaient sur les images d'entrée apparaissent en partie.

Enfin, au niveau de l'image segmentée(**Figure16**), on peut constater une forte ressemblance par rapport à l'image d'entrée(**Figure13**). Cependant, les objets

présentent des contours avec des zones d'ombre tout en conservant la forme du départ.

En conclusion, après avoir réalisé notre chaîne de traitement complète et l'appliquer sur les 4 images(objet1.jpg, objet2.jpg, objet3.jpg et objet4.jpg), l'analyse des résultats nous permet de déduire que, malgré l'application du même traitement sur les images, tous les résultats n'ont pas été similaires. Cela nous aide à comprendre aussi qu'autant que l'objet sur l'image de départ se distingue du fond de l'image, autant que les traitements s'applique avec efficacité, et cela permet de conserver la valeur des pixels et aussi la forme de l'image du départ.