Traitement d'Images Numériques :une introduction avec le logiciel ImageJ et le langage Java

La couleur et la ligne

Q1. Sur ***Clown***, commencer par utiliser ***Image/Type/RGBStack***. Qu'est-ce que cela fait ?

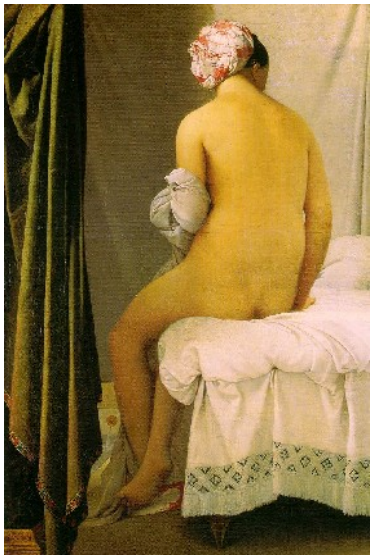
Les Stacks sont des piles d'images. Cela permet d'automatiser des traitements identiques sur des stocks d'images (*Batch mode*). Ou bien de faire de la pseudo visualisation 3D, à la manière d'un *gif* animé. Ou bien de travailler sur des séquences vidéos. Bref c'est un mode de représentation de vos images puissant.

Q2. Essayez de ne récupérer que les teintes rouges sur l'image de Clown.

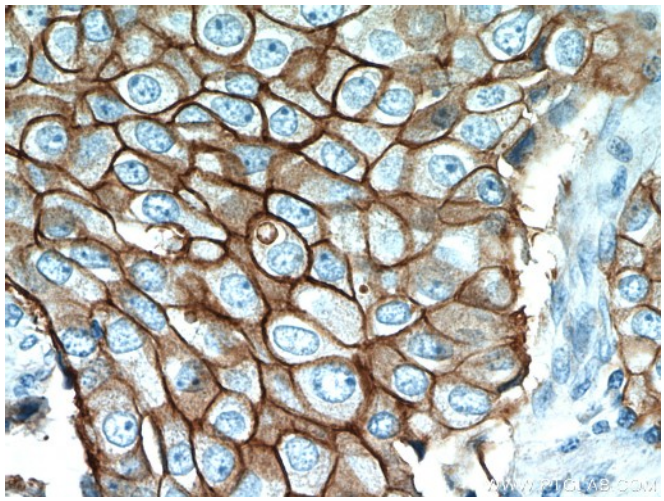
Q3. Récupérez les classes de plugins ***ColorSpace.zip***. Les installer chez vous. Et les appliquez à Clown. Notamment passez à l'espace *YCbCr*. *Y* capture la luminance et *Cr* et *Cb* capturent de l'information chromatique. Puis convertir le Stack obtenu en 3 images correspondant aux trois composantes couleurs Y, Cr et Cb. Convertissez également l'image initiale de Mandrill couleur en trois images correspondants aux composantes R, G et B. Comparez les décomposition.

Q4. **Ingres contre Delacroix.**

Utilisez ce que l'on a vu jusqu'à présent (travail sur les couleurs, les contours) pour vérifier le travail différent de ces deux peintres l'un plus sur la couleur et l'autre sur la ligne. Éventuellement quantifier cela.



Q5. Histopathologie numérique: quantification d'un biomarqueur type HER2/Neu



http://biogenex.com/us/c-erbb-2-her-2-neu-2333.html
http://www.ptglab.com/Products/ERBB2,p185-Specific-Antibody-18299-1-AP.htm

L'idée de cet exercice est lié au problème de découverte de nouvelles molécules pharmacologiques par la quantification de bio-marqueurs image en lien avec un traitement pharmacologique et éventuellement une analyse génotype-phénotype de la maladie et de son pronostic. On travaille sur l'image *HER2.jpg*.

a. Déconvolution couleur : ***Image/Color/Colour_Deconvolution*** sous Fiji (ou http://www.mecourse.com/landinig/software/cdeconv/cdeconv.html)

b. Quantification du marquage noyau (bleu).

c. Quantification du marquage membrane cellulaire (marron intense aux interfaces).

La géométrie : Ligne de Partage des Eaux et Watershed

On va travailler bien sûr sur l'image de ***Blobs.gif***. Vous conserverez les images résultats successives pour les comparer après.

Q1. Binarisez l'image (il faut une image codée sur 8 bits avec 2 couleurs 0 et 255).
Soit A l'image binaire avec les blobs en noir et B l'image inverse de A.
Appliquez la squelettisation sur B. Quelle structure obtient-on ?
Appliquez le Watershed binaire sur A. Quelle structure observe-t-on ?
Appliquez le Watershed binaire sur B. Quelle structure observe-t-on ?

Q2. Utilisez le plugin **Watershed.zip** fourni fonctionnant sur des images de Niveaux de Gris pour délimiter les zones d'influences des cellules de l'image ***Blobs.gif***.
Bien sûr, votre image doit être correctement pré-traitée auparavant (par des lissages successifs par exemple).
Combien de cellules accolées arrivez-vous à discriminer dans un premier temps? En sélectionnant l'option Animation, vous étudierez le déroulement de l'algorithme.

Q3. Marquez l'image de *Blobs.gif* de sorte à aider l'algorithme de Ligne de partage des Eaux (LPE) à discriminer la plupart des cellules accolées. Comparez l'évolution de l'algorithme sur l'image de la question 2 et celle marquée de cette question, en utilisant l'option Animation.

Q4. On va essayer autre chose. Récupérez la carte de Distance de l'image A. Normalisez au niveau de l'histogramme pour y voir quelque chose. Appliquez le Watershed en niveau de gris. Observez. Commentez.

Q5. Récupérez le code d'un ancien étudiant : **http://rsb.info.nih.gov/ij/plugins/watershed.html**. Cela peut toujours servir. Essayer à l'aide des Animations précédentes et éventuellement du code ainsi récupéré, de comprendre dans les grandes lignes de l'algorithme utilisé (voir Annexe).

Q12. A l'aide de l'archive **delaunay.jar**, faites le lien expérimental entre SKIZ, Triangulation de Delaunay, Diagramme de Voronoï et Watershed. **(Remarque** ! vtk.jar et java3d sont des archives compatibles avec imageJ pour faire de la géométrie algorithmique ou de la 3D mais cela dépasse le cadre de cette initiation).

Annexe :

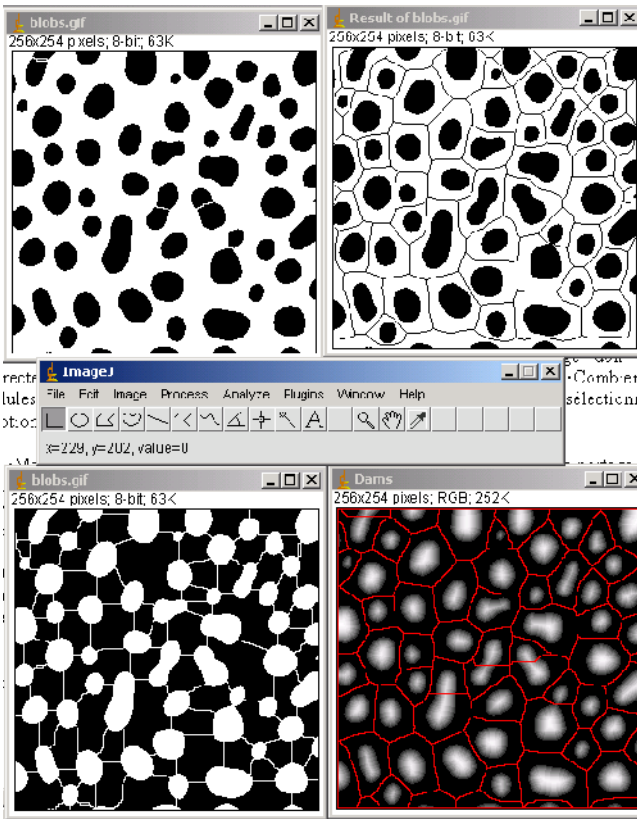


Illustration de l'animation de l'algorithme de Watershed

Des liens intéressants/importants :

http://fiji.sc/Developing_ImageJ_in_Eclipse

http://fiji.sc/Source_code

http://wiki.linux-france.org/wiki/Les_commandes_fondamentales_de_Linux

D'autres outils :

http://www.cellprofiler.org/examples.shtml

http://research.mssm.edu/cnic/tools-ns.html

http://www.cbi-tmhs.org/Dcelliq/downloading.html

Exercice de programmation en Java : Simuler une image.

Maintenant que vous savez analyser et quantifier une image, sauriez-vous programmer en java ce plugin de imageJ qui construit des biomorphes autrement dit des images fractales. Voici la règle de construction ou pseudo-algorithme pour la :

Création des biomorphes

Noir et blanc:

Pour chaque point du plan imaginaire (z= x + i y)
On lance une boucle de 10 itérations.
on calcule z=f(z)
si(|x| >10 ou |y| > 10 ou |z| >10)
on quitte la boucle.
En fin de boucle :
si (|x|>10 ou |y|>10)
on marque un pixel noir sur fond blanc

Couleur:

La couleur dépend du nombre d'itérations et de la valeur de
|x|,|y|,|z|.

Les biomorphes sont du type: http://fr.wikipedia.org/wiki/Fractale

$F(z)=z^{Exposant}+Cst.$

Essayez les ensembles de Julia suivant :

c=0.3+0.6i c=-0.0519+0.688i c=0.32+0.43i c=-0.0986-0.65186i c=0+i	c=-0,4+0,7i c=-0,7+0,3i c=-1,77+0,01i c=-0,15+0,45i