

TP TI, semaine 10: Détection de contours par approches du second ordre

François LEPAN

5 avril 2013

Introduction

Dans ce rapport nous allons voir comment faire de la détection de contour par approche du second ordre (Laplacien). Le Laplacien, qui est la dérivé seconde d'une fonction image, permet d'identifier les contours d'objet dans une image. Nous utiliserons l' image sur la Fig.1 pour mettre en évidence les étapes de la détection de contour.

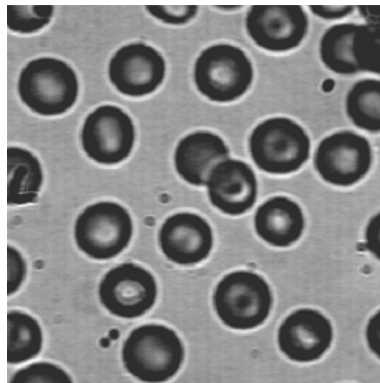


FIGURE 1 – L'image Spores

1 Calcul du Laplacien

Après avoir exécuté le plugin sur l'image spores (*cf.* Fig.1) on obtient la Fig. 2.

On observe sur ce résultat que pour chaque spore il y a plusieurs couleurs qui reviennent. Chaque contour a une couleur proche du noir ensuite du blanc et on revient sur la couleur grise du fond. Ceci est dû au calcul du Laplacien. En effet lorsque l'on calcule la dérivé seconde de la fonction image les contours sont marqués par le passage par zéro des valeurs des pixels qui donne cette couleur noire (*cf.* Fig.3).

Afin de mieux visualiser les passages par 0 nous allons seuiller les valeurs des pixels.

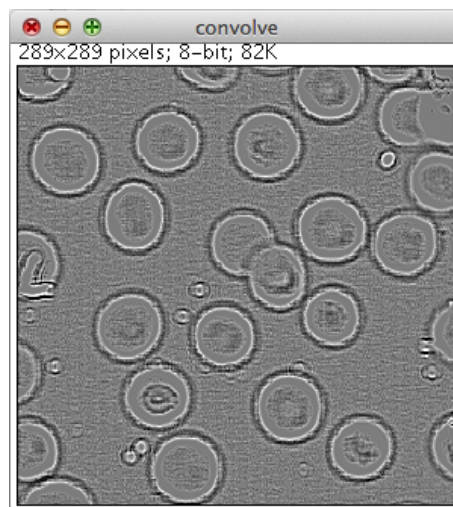


FIGURE 2 – Résultat du calcul de la convolution du Laplacien

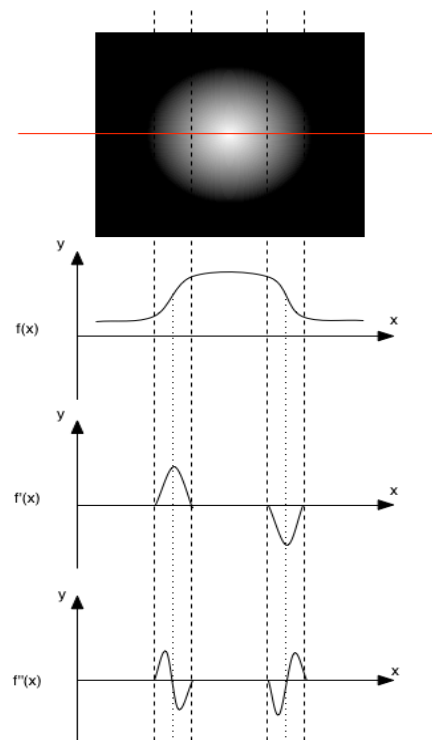


FIGURE 3 – Courbes de dérivations d'une fonction image

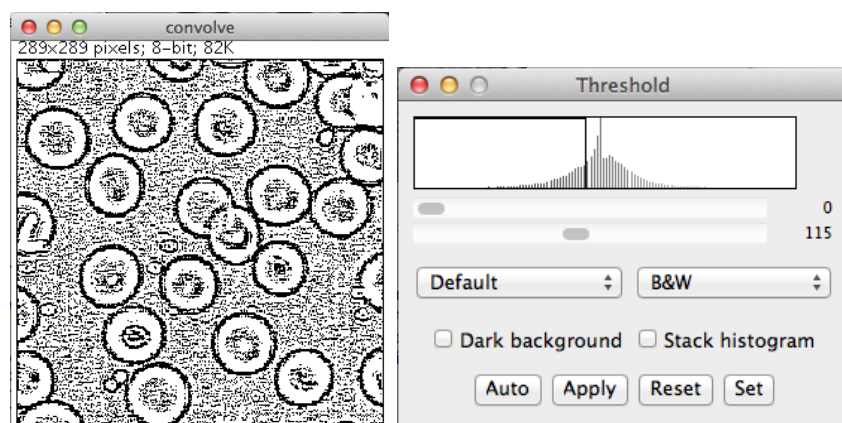


FIGURE 4 – Résultat du calcul de la convolution du Laplacien seuillé pour mettre en évidence les pixels auxquels le Laplacien a une valeur proche de 0

On observe sur la Fig.4 la Fig. 2 seuillée pour mettre en évidence les pixels auxquels le Laplacien a une valeur proche de 0.

Malgré un seuillage des valeurs on voit sur les bords des spores que le contour n'est pas uniforme. Ceci est dû au fait que certaines valeurs ne passent pas exactement par la valeur zéro.

Afin de résoudre ce problème nous allons faire un seuillage manuel des passages par 0 du Laplacien.

2 Seuillage manuel des passages par 0 du Laplacien

Le seuillage manuel consiste à mettre en évidence le passage par zéro des valeurs des pixels. On prend une image noire et on met en blanc les pixels considérés comme contour.

L'algorithme appliqué est le suivant :

On choisit un seuil (ici 15).

On crée une nouvelle image qui va contenir les passages par zéro.

On choisit le voisinage du point courant (nous prendrons 3x3)

Pour chaque pixels :

On vérifie si dans le voisinage il y a bien changement de signe.

Si il y a changement et que les valeurs sont significatives
on met au même emplacement que le pixel courant du Laplacien
la valeur 255 dans la nouvelle image.

En appliquant cet algorithme sur la Fig.2 on obtient la Fig.5. Les contours obtenus sont ici beaucoup plus fins et nets que pour un seuillage simple. Mais pour certain spore, le contour n'est pas fini et certains pixels ont été choisis comme contour alors qu'ils ne le sont pas. Ceci est dû au calcul de la dérivation qui génère du bruit.

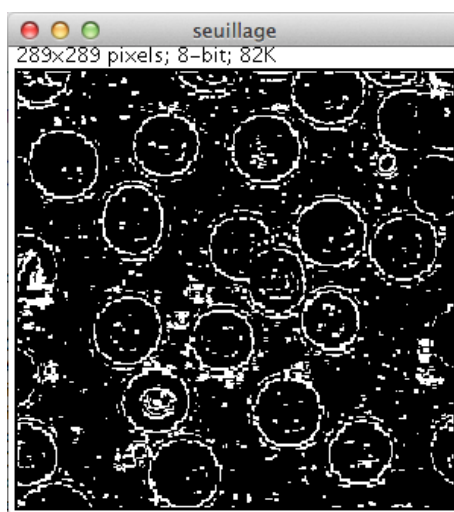


FIGURE 5 – Seuillage manuel des passages par 0 du Laplacien

Une solution pour atténuer ce bruit est d'appliquer un filtre gaussien avant l'application du Laplacien qui va lisser l'image et apporter les corrections nécessaires à la détection de contour.

3 Utilisation du filtre LoG

Tout d'abord il faut générer un masque. Pour se faire une fonction *masqueLoG(int tailleMasque, float sigma)* existant permet de le créer. Mais comment choisir les valeurs de la taille du masque et celles du σ ?

σ correspond à l'écart-type de la gaussienne c'est à dire l'étalement de la cloche. Plus l'étalement est grand plus le filtre doit l'être. Il faut donc choisir une constante suffisamment grande pour que la taille du filtre soit supérieure ou égale à la cloche de la gaussienne.

On aura donc $\text{tailleMasque} \geq K * \sigma$ avec *tailleMasque* impaire. Nous choisirons $K = 6$.

Après quelques tests pour trouver σ et le bon seuil nous trouvons $\sigma = 1.4$ et le seuil à 6. Après exécution on obtient la Fig. 6.

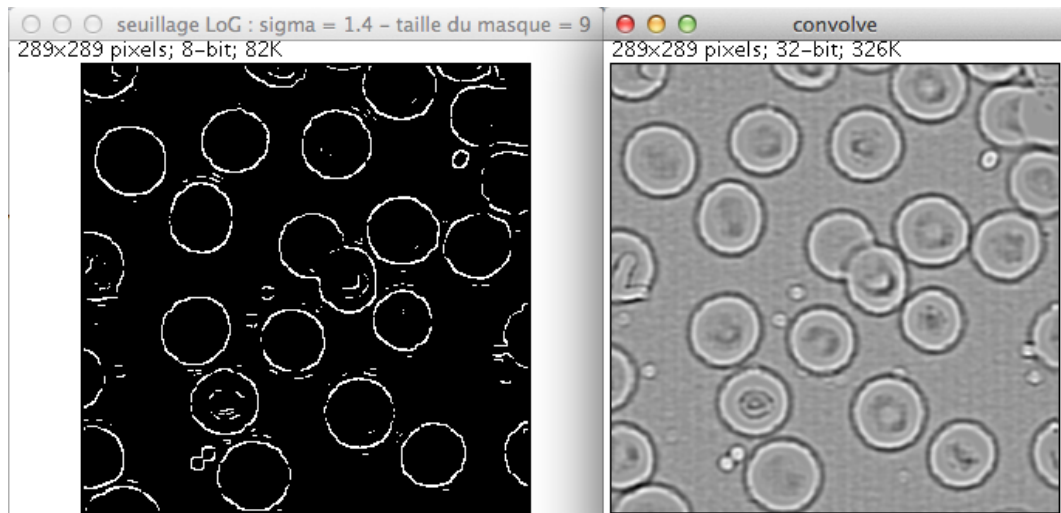


FIGURE 6 – À gauche le seuillage manuel du LoG (Laplacian of Gauss) avec un seuil = 6 et à droite la convolution du Laplacien lissée par un filtre gaussien avec $\sigma = 1.4$

On voit que le LoG donne de meilleurs résultats que le Laplacien seul. Il y a beaucoup moins de bruit et donc on distingue plus les contours.

Conclusion

Nous venons de voir la détection de contour par approche du second ordre. On constate que la détection est bien faite mais encore imparfaite. Mais si on la compare à une approche du première ordre la réduction de contour indésirable est plus importante.