COMPTE RENDU

TP10 : DETECTION DE COUNTOURS PAS APPROCHES DU SECOND ORDRE

Par
FRANCOIS Rémy & MIRANDA Yoan

INTRODUCTION

Le Laplacien est un masque qui permet de repérer les contours de motifs présents dans une image. En effet, le Laplacien correspond à la dérivée seconde, les passages à 0 du Laplacien correspondent donc aux contours de l'image.

CALCUL DU LAPLACIEN

```
fpLaplacian.convolve(MASQUES_LAPLACIENS3x3[filtre], 3, 3);
this.imp = new ImagePlus(titre+"_exo1"+extension, fpLaplacian);
this.imp.show();
```

Ces 3 instructions permettent d'afficher l'image résultant du calcul du Laplacien.
En effet, la première instruction permet de calculer les valeurs du Laplacien grâce à un masque de convolution. Puis, on crée une image correspondant aux valeurs du Laplacien pour ensuite l'afficher.

Apres avoir appliqué le plugin sur l'image spores et ajustés le contraste de l'image pour avoir des valeurs reparties entre 0 et 255 pour augmenter la luminosité de l'image et donc mieux permettre de voir les contours, on obtient les images suivantes :

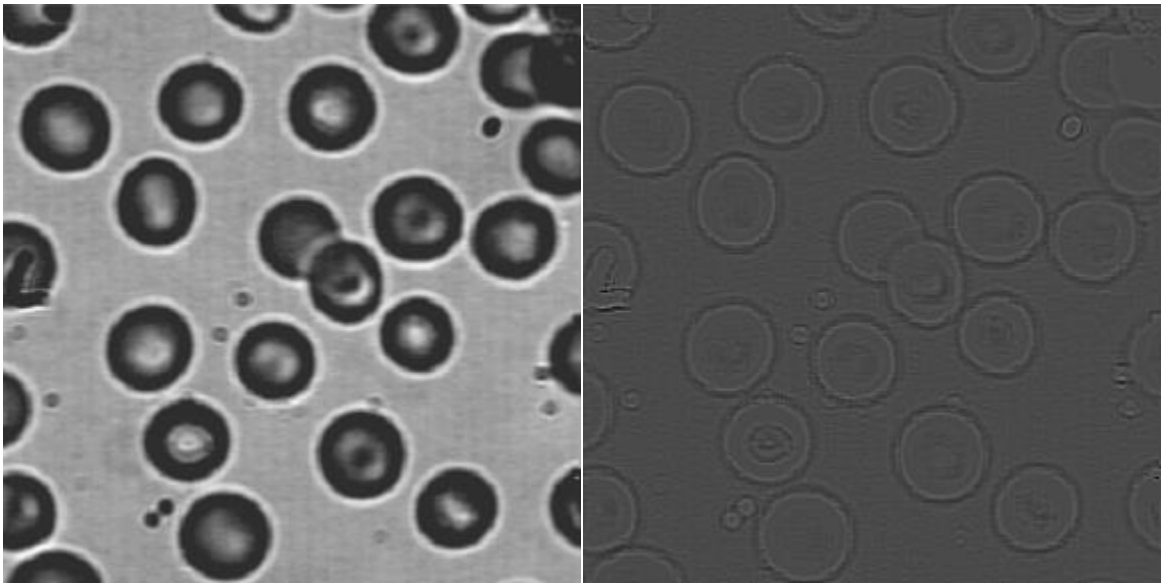

Image originale (à gauche) et image après calcul du Laplacien (à droite)

On observe donc sur cette image les contours des motifs. Le fond de l'image est en gris, les contours extérieurs en noirs et les contours intérieurs en blanc. Ces couleurs correspondent respectivement aux valeurs initiales nulles, négatives et positives de la dérivée seconde.

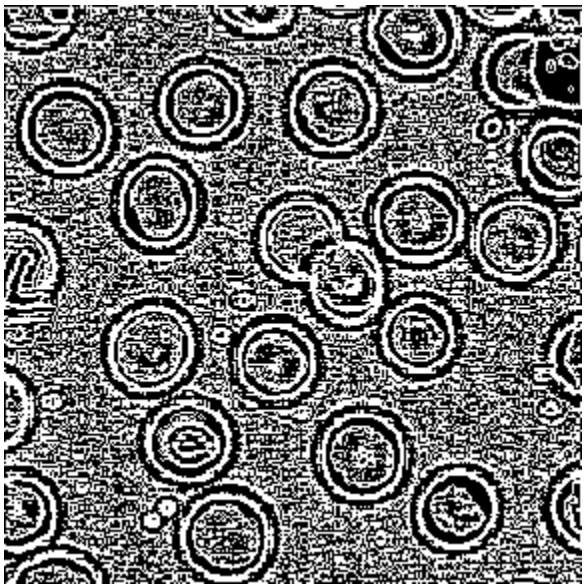

Image obtenue après mise en évidence des pixels
auxquels le Laplacien à une valeur proche de 0

En sélectionnant uniquement les pixels ayant une valeur comprise entre -1,36 et 0,6, on observe que certains contours sont tronqués. Cela s'explique par le fait que le Laplacien ne passe pas toujours par 0. Il faut donc seuiller les passages par 0 pour avoir un résultat plus correct.

SEUILLAGE DES PASSAGES PAR 0 DU LAPLACIEN

Pour avoir l'image binaire du passage en 0 du Laplacien selon un seuil, on parcourt les pixels de l'image. Pour chaque pixel, on récupère la plus petite valeur et la plus grande valeur du voisinage. Si la valeur minimale est inférieure à -seuil et la valeur maximale supérieure à +seuil, cela veut dire qu'il y a eu un changement de signe. Le pixel courant est donc un pixel de contour.

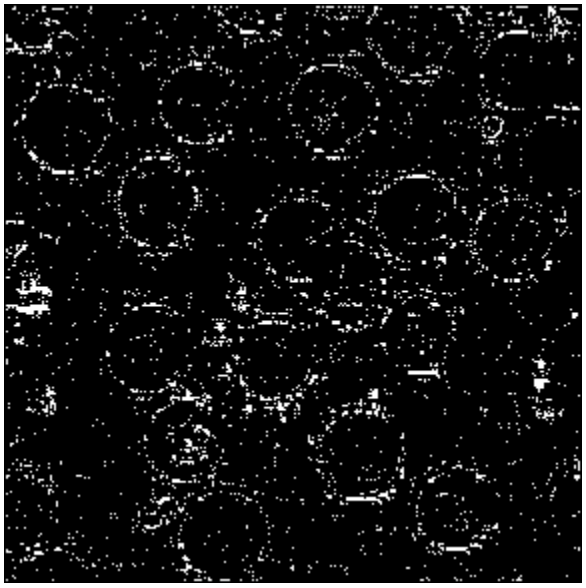

Image binaire des passages par 0 obtenue

En choisissant un seuil de 15, on observe plus précisément les contours. Toutefois, on observe que l'image est bruitée. On va donc appliquer un filtre gaussien pour filtrer l'image et mieux voir les contours.

UTILISATION DU FILTRE LOG ET DETECTION MULTI-ECHELLES

Sigma correspond à l'écart-type de la loi gaussienne. Cela veut dire que plus sigma est élevée, plus la loi gaussienne est repartie, il faut donc que la taille du filtre augmente lorsque σ augmente pour que toutes les valeurs non nulles de la loi gaussienne soient présentes dans le filtre. On doit donc avoir tailleMasque >= 6*σ pour retenir plus de 99% des valeurs de loi gaussienne.

Apres avoir utilisé la loi gaussienne avec un σ = 3, nous obtenons l'image suivante après ajustement des valeurs de l'image pour avoir des valeurs comprises entre 0 et 255.

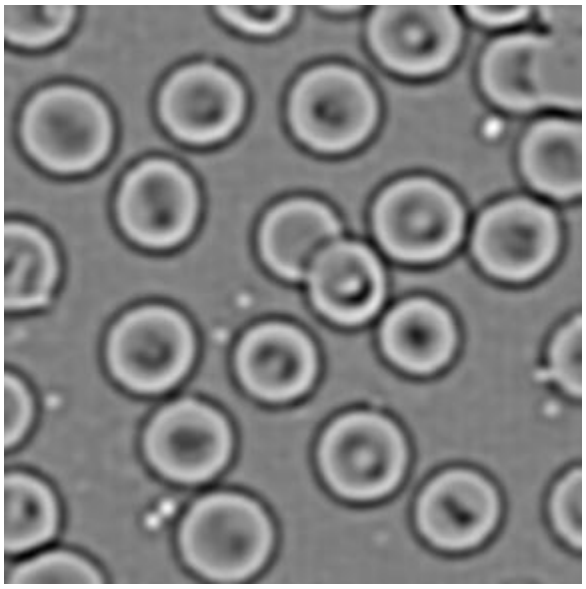

Image obtenue après utilisation de loi gaussienne avec un σ de 3

Si nous utilisons un seuil de 1 pour binariser l'image selon les passages en 0 du Laplacien comme dans l'exercice précèdent, nous obtenons l'image suivante :

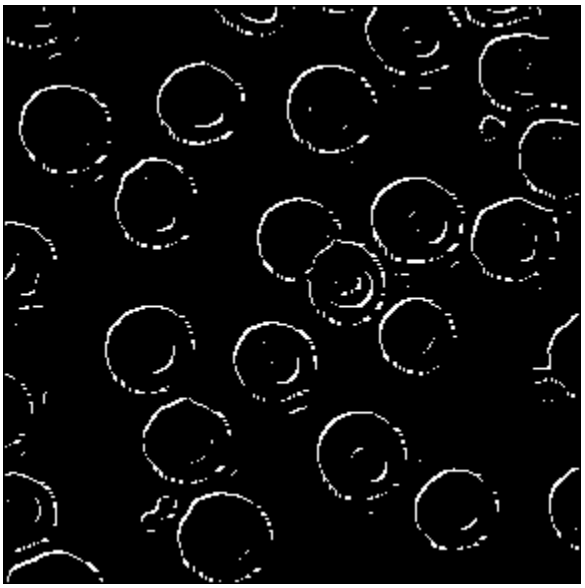

Image binarisée selon les passages en 0 avec un seuil de 1

On remarque donc que le LoG a permis de filtrer l'image et par conséquent de réduire le bruit. Les pixels blancs représentent désormais uniquement des pixels de contour.

CONCLUSION

Pour conclure, ce TP nous a permis de manipuler le Laplacien, qui permet de détecter les contours des objets présents dans la scène. Au début nous avions un résultat qui était peu net, avec beaucoup d'éléments parasites. Mais nous avons pu éliminer ces parasites en seuillant les passages par 0 du Laplacien puis en utilisant le Laplacien de gaussienne pour à la fin obtenir une image binarisée beaucoup plus nette que celle obtenue au départ.