

# TI : TP 11 : Détection des contours par approches du second ordre

**binôme** : Benjamin Ruytoor et Aurore Allart

**date** : 29 mars 2013

## Introduction

Le calcul du Laplacien, qui est en fait la dérivée seconde d'une fonction image, nous permet de détecter avec plus de précision les contours qu'avec l'approche des dérivées premières. Cette méthode fournit, après calcul, des réseaux de lignes fermées. De plus, cette méthode est très proche de la façon que l'œil voit les contours des objets. Cependant cela demande plus de calculs. Le Laplacien est très sensible au bruit, ce qui demande des opérations sur l'image supplémentaire comme le lissage.

Dans ce TP, nous allons coder le Laplacien, ainsi que le seuillage et l'opérateur LoG (Laplacian of Gaussian) dans un plugin qui nous permettra d'utiliser ses fonctions sur des images ouvert avec imageJ.

## 1. Calcul du Laplacien

Le calcul du Laplacien nous permet de restituer les contours des objets d'une image. Malheureusement ce calcul apporte beaucoup de bruit (pixel non désirable), et des contours épais, et donc on ne peut utiliser cette image.

Des traitements supplémentaires sont nécessaires pour enlever les bruits (par un filtre par exemple) et réduire l'épaisseur des contours. Voici l'image obtenue, nous avons du ajuster la brillance/contraste de l'image pour voir ce que l'image contient (de base elle était très sombre) :

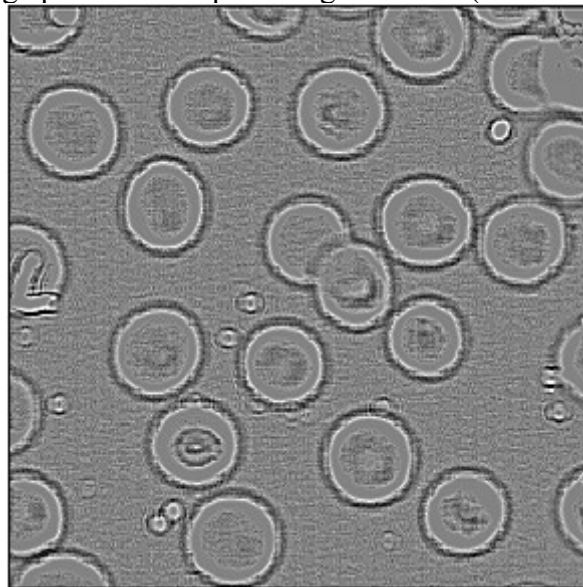


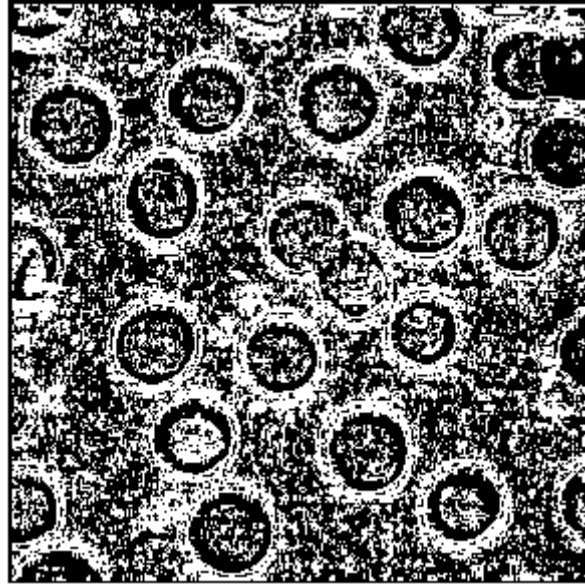
Figure 1 : calcul du Laplacien.

## 2. Seuillage des passages par 0 du Laplacien

Le seuillage des passages par 0 permet de restituer une image en noir et blanc. Le passage par zéro permet de détecter les points contours des objets.

### 1. Seuillage manuel des passages par 0 du Laplacien

Le seuillage manuel, c'est à dire qu'on lui donne la valeur du seuil. Pour l'image des spores, le seuil est de 15. Malheureusement, les points contours ne sont pas très bien définis : les bruits restent nombreux.



**Figure 2** : seuillage manuel par passage à zéro.

## **2. Seuillage automatique des passages par 0 du Laplacien**

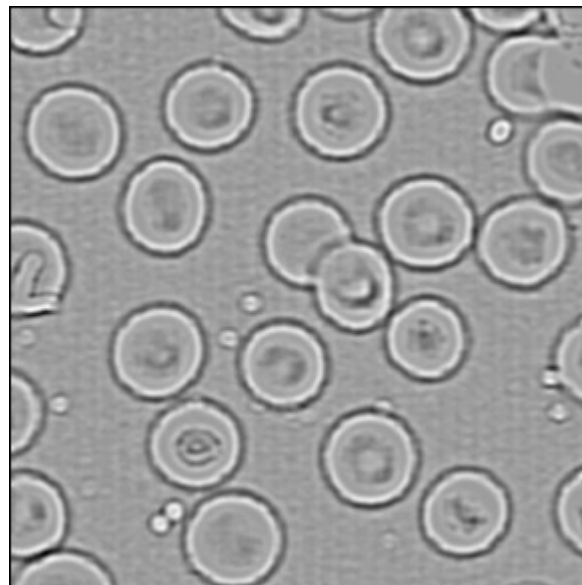
Pour le seuillage automatique, il faut prendre la taille de l'histogramme et

## **3. Opérateur LoG et détection multi-échelles**

LoG permet de lisser l'image et donc de réduire le bruit en pré-filtrant avec le noyau gaussien avant d'utiliser le Laplacien.

### **1. Génération de masques pour l'opérateur LoG**

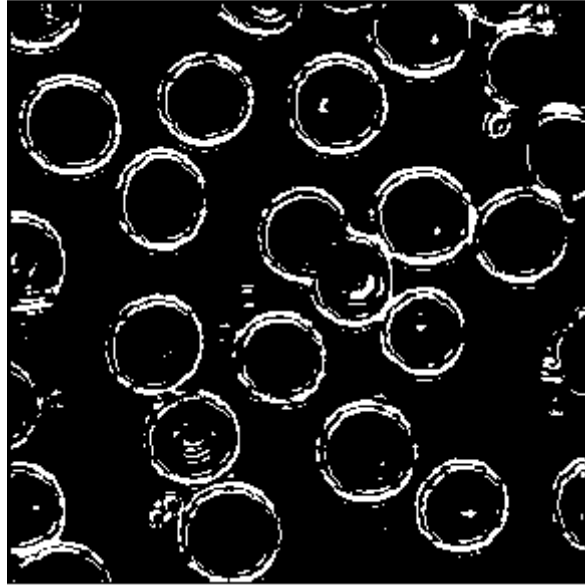
Un lien existe entre le sigma et la taille du masque, plus sigma est grand et plus la taille du masque. On a trouvé que la taille du masque est égal 7 fois sigma. De plus, la taille du masque doit obligatoirement être impaire car sinon, c'est difficile de définir le centre du masque. De plus, on constate qu'il y a un lien entre le seuil et sigma : plus sigma est grand, et plus le seuil est petit. Voici le résultat du masque avec un sigma égal à 1.4, ce qui nous donne, un masque 9\*9.



**Figure 3** : Utilisation du masque 9\*9 sur l'image de départ.

On constate par rapport à la figure 1, que l'image contient des couleurs plus lisses et donc des contours plus lisses. Quand on seuille manuellement à 11 les passages par 0 de cette image, on

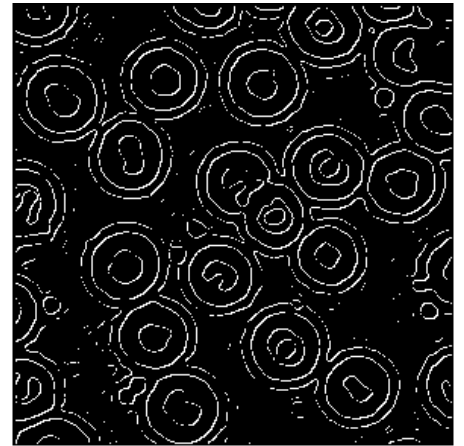
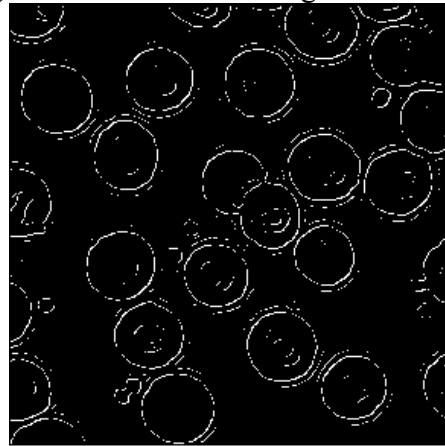
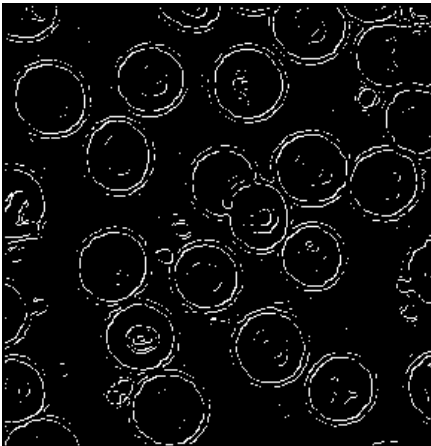
obtient les contours des spores. Malheureusement, on constate des trous dans ces contours. En voici l'image :



**Figure 4** : seuillage du masque LoG.

## 2. Détection multi-échelles

Voici les différentes images avec les différents sigma :



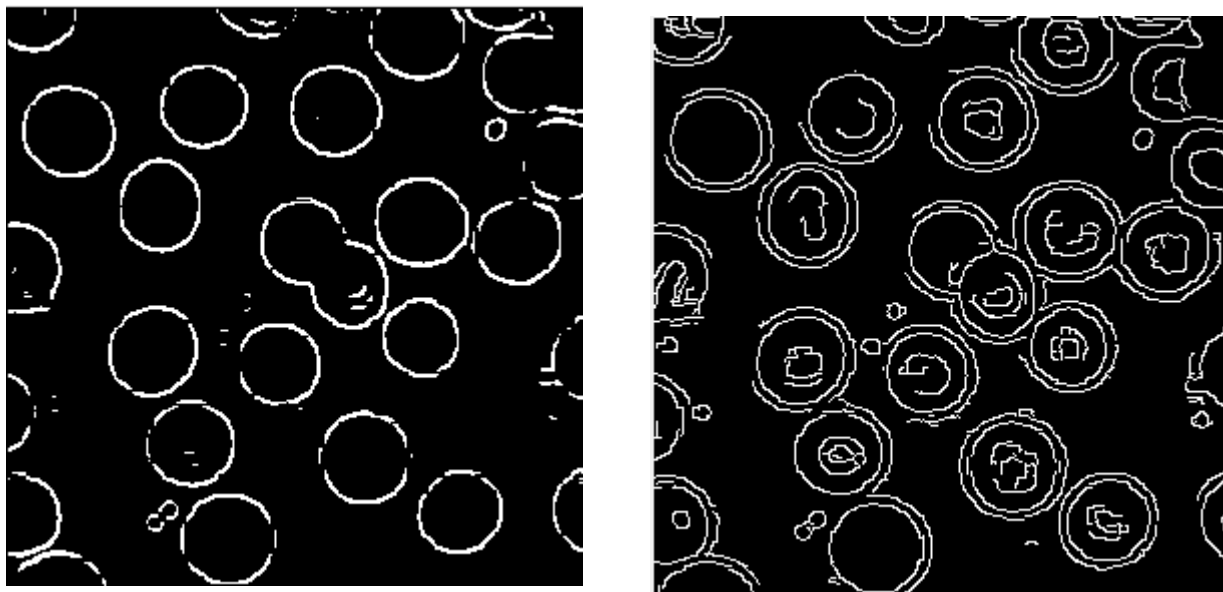
**Figure de gauche** : seuil avec un sigma de 1.4.

**Figure du centre** : seuil avec un sigma de 2.

**Figure de droite** : seuil avec un sigma de 3.

On constate que plus, le seuil est haut, et plus les contours sont fermés.

### 3. Utilisation de la norme du gradient



**Figure de gauche :** image calculée le plugin gradient.

**Figure de droite :** image calculée avec imageJ.

On constate que dans la figure de gauche, les contours sont épais et non fermés, ce qui est différent avec l'image de droite.

### Conclusion

L'implémentation des gradients pour trouver les contours est plus facile malheureusement le rendu est grossier : trop de bruit, contours ouvert, par contre les calculs sont rapides. Là est la différence avec le Laplacien, cependant les calculs sont beaucoup plus important.