

# TP IMG : détection de contours

---

## Author

---

- Selim Lakhdar [selim.lakhdar@gmail.com](mailto:selim.lakhdar@gmail.com) [selim.lakhdar.etu@univ-lille.fr](mailto:selim.lakhdar.etu@univ-lille.fr)

## Seuillage de la norme d'un gradient

---

### Comparaison avec le résultat fourni par ImageJ

- On utilise une profondeur de 32bits car on manipule des valeurs qui peuvent être négatives.
- min:0 max:255. La proportion de pxl noir/blanc est plus importante à cause du filtre.
- Avec une profondeur de 32 bits l'image est (presque) toute blanche. En assombrissant l'image les contours apparaissent. (En utilisant l'option "Auto")
- Numériquement, quand on soustrait les deux images, on obtient un fond noir. (images identiques)

### Seuillage de la norme du gradient précédemment calculée

- Pas facilement
- macro2.txt
- Oui, mais il reste des détails qui n'apparaissent pas encore.

## Détection des maxima locaux de la norme d'un gradient

---

### Calcul de la direction du gradient

- macro3.txt
- 

### Détection des maxima locaux

- je ne suis pas arrivé à obtenir la même image. Le problème doit sûrement venir dans mon parcours de g\_img (Ou la conversion du gradient en angle ???)

## Seuillage des maxima locaux par hystérésis

---

```

public ByteProcessor hystIter(ImageProcessor imNormeG, int seuilBas, int seuilHaut) {

...

// => Cette noucle ne va rien faire tq on ne trouve pas de pxl > seuilBas.
// Si le seuilHaut est dépassé elle prendra la val max (255).
// Sinon,128.

    for (int y=0; y<height; y++) {
        for (int x=0; x<width; x++) {

            int g = imNormeG.getPixel(x, y)&0xFF;
            if (g<seuilBas) continue;

            if (g>seuilHaut) {
                maxLoc.set(x,y,255);
                highpixels.add(new int[]{x,y});
                continue;
            }

            maxLoc.set(x,y,128);
        }
    }

    int[] dx8 = new int[] {-1, 0, 1,-1, 1,-1, 0, 1}; // Filtre/masque/drv pour x
    int[] dy8 = new int[] {-1,-1,-1, 0, 0, 1, 1, 1}; // Filtre/masque/drv pour y

    while(!highpixels.isEmpty()) {

        List<int[]> newhighpixels = new ArrayList<int[]>();

        for(int[] pixel : highpixels) {
            int x=pixel[0], y=pixel[1];

            for(int k=0;k<8;k++) {
                int xk=x+dx8[k], yk=y+dy8[k];
                if (xk<0 || xk>=width) continue; // on saute les bords
                if (yk<0 || yk>=height) continue; // on saute les bords
                if (maxLoc.get(xk, yk)==128) { // on traite que les pxl de val 128
                    maxLoc.set(xk, yk, 255);
                    newhighpixels.add(new int[]{xk, yk});
                }
            }
        }

        highpixels = newhighpixels;
    }

    for (int y=0; y<height; y++) {
        for (int x=0; x<width; x++) {
            if (maxLoc.get(x, y)!=255) maxLoc.set(x,y,0);
        }
    }

    return maxLoc;
}

```