

# Traitement d'images

## TP 3

Garcia Maxime, Palisse Benjamin, Petitbon Pierre

L'objectif de ce TP est de comparer deux méthodes de détection de contour d'une image, l'une par gradient l'autre par laplacien. Nous testerons les deux programmes sur des versions plus ou moins bruitées d'une image dont les contours intérieurs et extérieurs sont représentés sur les figures suivantes :

FIGURE 1 – contour externe

FIGURE 2 – contour interne

Nous testerons aussi nos programmes sur les images *tangram.pgm* et *lacornou.pgm*, cette dernière étant bien plus compliqué.

FIGURE 3 – tangram.pgm

FIGURE 4 – lacornou.pgm

## 1 Détection par gradient

### 1.1 gradientNaif

Cette détection de contour est obtenu en calculant le gradient de notre image, puis en considérant que les contours sont les points où le gradient est le plus important sans utiliser la suppression des non-maxima. En appliquant cette méthode sur l'image bruitée suivante (image simulant un bruit multiplicatif de type speckle), on obtient ceci :

FIGURE 5 – image ayant un bruit multiplicatif de type speckle

FIGURE 6 – image ayant un bruit multiplicatif de type speckle

FIGURE 7 – image ayant un bruit multiplicatif de type speckle

On constate qu'en identifiant les points ayant un gradient supérieur à 250 comme point de contour, le programme identifie bien la forme principale des différents objets de l'image mais il interprète l'intérieur des objets aussi comme des contours. En augmentant la valeur du seuil à 680, on constate que le seuil est trop grand car même les points qui sont réellement des contours tendent à disparaître. En augmentant encore le seuil, c'est la totalité de la forme qui disparaît. Cette méthode n'est donc pas bonne car le programme n'est pas capable de distinguer le contour de l'intérieur de l'objet.

## 1.2 gradient Prewitt

Cette fois-ci, on supprime les non maxima et on fait un seuillage par hystérésis en utilisant le gradient de Prewitt.

## 2 Détection par Laplacien

On utilise la détection de contour sur l'image *formes1bb25.pgm*. Si l'on prend  $\sigma_1 = 10$  et  $\sigma_2 = 6.25$  (le rapport des 2 faisant bien 1.6), on obtient

FIGURE 8 – laplacien avec 10 et 6.25 pour valeur de filtre

Les formes principales des contours ressortent, cependant le programme rajoute de nombreux contours "parasites" autour des formes à cause du bruit. Les contours reconnus par le programme sont plus arrondis que ne le sont les originaux réellement.

Avec comme paramètre des filtres lors de la différence de gaussienne  $\sigma_1 = 30$  et  $\sigma_2 = 18.75$ , on obtient les contours suivants :

Les contours sont très apparents mais plus arrondis que l'image originale, cela est notamment très visible sur le rectangle en bas. Les traits un peu "tremblants" de la figure

FIGURE 9 – laplacien avec 30 et 18.75 pour valeur de filtre

au milieu en haut dans l'image initiale apparaissent ici très courbés. La figure de droite est devenu un rond. Par rapport au cas précédent, les contours sont plus arrondis, et donc moins fidèle à la réalité, mais les contours superflus des bords ont disparu.

Pour l'image tangram, avec  $\sigma_1 = 5$  et  $\sigma_2 = 3.125$  :

FIGURE 10 – laplacien avec 5 et 3.125 pour valeur de filtre

Les triangles initiaux sont reconnus par le programme, ainsi que le petit effet de relief qu'il y a en chacun d'eux. Cependant, de nombreux contours d'objets qui n'étaient pas dans l'image initiale apparaissent, comme dans les cas précédents. Le coté gauche à l'origine vide est alors remplie par des objets.

Si l'on augmente les valeurs des filtres,  $\sigma_1 = 12$  et  $\sigma_2 = 7.5$ , on obtient :

FIGURE 11 – laplacien avec 12 et 7.5 pour valeur de filtre

Les triangles de la figure initiale apparaissent aussi dans le résultat produit par notre algorithme. Cependant, ils prennent une forme plus ronde sur les bords, et l'effet de relief disparaît. Les contours apparaissant sur les bords du aux bruits sont moins nombreux.

Pour l'image lacornou, en prenant  $\sigma_1 = 8$  et  $\sigma_2 = 5$

FIGURE 12 – laplacien avec 8 et 5 pour valeur de filtre

On constate que pour une image avec ce niveau de détail, notre programme ne fournit pas des contours satisfaisant.