

# Traitement d'images TP2

Ahmed Amine DAOU      Wrushabh Warshe  
[ahmed.amine.daou@umontreal.ca](mailto:ahmed.amine.daou@umontreal.ca)      [wrushabh.warshe@umontreal.ca](mailto:wrushabh.warshe@umontreal.ca)

October 29, 2018

# Introduction

## Description du but de TP

Ce qui est demandé dans le cadre de ce TP2 de traitement d'images est d'implémenter un algorithme de détection de contour (Canny) basé sur des notions vues en cours telles que le filtrage, calcul de gradient, suppression des non maximums et le seuillage par Hystérésis. Une autre implementation Semi-Auto de Canny est demandée. Ce rapport montrera des résultats expérimentaux à chaque étape de l'implémentation.

## Détection et rôle des contours dans une image

L'existence des contours sert à reconnaître les objets, distinguer les zones présentes, faire de la segmentation et extraire des informations caractérisant l'image. Un contour se manifeste par une repture d'intensité dans une direction. Pour détecter cette rupture on utilise généralement (1) un calcul de gradient dans une direction choisie puis (2) effectuer un seuillage permettant de mettre en évidence des contours fins et sans bruits.

## Filtre gaussien

Dans l'algorithme de Canny, la première étape à effectuer est l'application d'un filtre gaussien (passe-bas) qui permet de réduire le bruit sur l'image originale avant de détecter les contours et d'éliminer les pixels isolés qui pourront avoir une norme de gradient élevée ce qui peut conduire à avoir de faux résultats.

$$\begin{aligned} \text{Gaussienne}[Image] &= \text{Gaussienne}[(x, y)], \forall (x, y) \\ &= \exp\left(\frac{-\pi(x^2 + y^2)}{\sigma^2}\right), \forall (x, y), \sigma = 1 \text{ dans notre cas} \end{aligned}$$

## Méthode utilisée pour calculer le gradient, la norme et de l'angle

### Le gradient:

Soit  $f(x, y)$  la fonction de l'image sur laquelle on calculeras les gradients. Pour calculer le gradient en  $x$  et  $y$  le principe utilisé est l'approximation de gradient.

- En  $x$ :

$$\frac{\partial f}{\partial x} = \lim_{\Delta x \rightarrow 1} \frac{f(x + \Delta x, y) - f(x, y)}{\Delta x}$$

Ça revient à convoluer l'image par un masque  $\Rightarrow \begin{bmatrix} 1 & -1 \end{bmatrix}$  ou  $\begin{bmatrix} -1 & 1 \end{bmatrix}$

- En  $y$

$$\frac{\partial f}{\partial y} = \lim_{\Delta y \rightarrow 1} \frac{f(x, y + \Delta y) - f(x, y)}{\Delta y}$$

Ça revient à convoluer l'image par un masque  $\Rightarrow \begin{bmatrix} 1 \\ -1 \end{bmatrix}$  ou  $\begin{bmatrix} -1 \\ 1 \end{bmatrix}$

### La norme:

Soit  $Norme_{x,y}$  la norme du gradient aux points  $x, y$ . Pour chaque pixel de la matrice on applique la formule suivante:

$$Norme_{x,y} = \text{sqrt}(Gradient_x \times Gradient_x + Gradient_y \times Gradient_y)$$

Soit  $Angle_{x,y}$  l'angle de la normale au gradient aux points  $x, y$ . Pour chaque pixel de la matrice on applique la formule suivante:

$$Angle_{x,y} = \arctan\left(\frac{-Gradient_x}{Gradient_y}\right)$$

## Nombre de seuils dans le filtre de Canny

Le seuillage est un point-clé dans la détection des contours et pour cela on utilise deux seuils un haut  $\tau_h$  et un bas  $\tau_l$  qui seront comparés à la norme du gradient de chaque point. À chaque point la décision se fait suivant les critères suivants :

- Si  $Norme_{x,y} < \tau_l$ , le point  $(x,y)$  n'est pas pris en compte
- Si  $Norme_{x,y} > \tau_h$ , le point  $(x,y)$  est accepté comme contour;
- Si  $\tau_l < Norme_{x,y} < \tau_h$ , le point  $(x,y)$  est accepté s'il est connecté à un point déjà accepté.

## Méthode utilisée pour Calculer les seuils à partir de l'histogramme

L'histogramme de l'image sur laquelle on effectuera la détection de contours, représente la distribution des différentes valeurs des pixels de l'image de la norme du gradient. Le résultat obtenu de la fonction `compute_histo(NormeGradient)` est un histogramme normalisé (valeurs entre 0 et 1), à partir de cet histogramme on calcule l'histogramme cumulé. dans le cas de l'algorithme de canny semi-automatique on définit  $\tau_h$  en parcourant l'histogramme cumulé jusqu'à arriver à la valeur  $H_{cumul}[i] < P_h < 1$  où  $\tau_h = H_{cumul}[i] \times 100$

## Approximation de l'angle

La technique utilisée pour approximer les angles est expliquée ci-dessous:

$$Angle_{x,y} = \arctan\left(\frac{-Gradient_x}{Gradient_y}\right)$$

la fonction utilisée en C pour calculer l'arctangente est `atan()` de `math.h`

$$x \mapsto \text{atan}(x)$$

$$]-\infty, -\infty[ \mapsto \left] \frac{-\pi}{2}, \frac{\pi}{2} \right[$$

après avoir appelé la fonction `atan()`, on vérifie dans quel intervalle parmi ces 4 se trouve l'angle  $[-\frac{\pi}{2}, -\frac{\pi}{4}[$ ,  $[-\frac{\pi}{4}, 0[$ ,  $[0, \frac{\pi}{4}[$ ,  $[\frac{\pi}{4}, \frac{\pi}{2}[$  on calcule les différences entre l'angle et les deux bornes de son intervalle, l'angle approximé est la valeur de la borne la plus proche à l'angle.

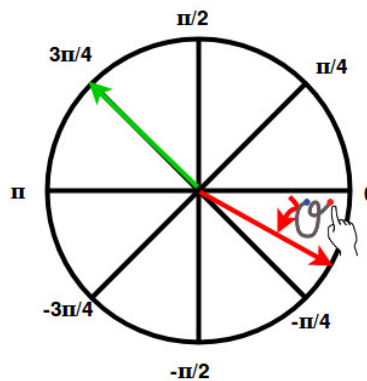


Figure 1: Exemple avec un angle négatif

Si l'angle  $\theta$  retourné par `atan()` est entre  $-\frac{\pi}{4}$  et 0 son estimation est de  $-\frac{\pi}{4}$  mais uniquement car l'estimation est négative on ajoute  $\pi$  à l'angle pour rester dans l'intervalle  $[0, \pi[$  ce qui donne une estimation de  $\frac{3\pi}{4}$ . D'où la valeur de  $\tau_l = \tau_h \times 0.5$

## But de la suppression des non-maximums dans le filtre de Canny

À un pixel précis si la norme du gradient est élevée, il y a une forte probabilité qu'il soit un contour. Mais la probabilité que ce point soit un contour n'est pas suffisante pour que la décision soit ferme (affirmer que c'est un contour ou non). Dans ce cas, pour être sûr on ne conserve que les maximums locaux (CAD. le considérer comme contour) et on supprime les non maximums et on repete l'opération sur les autres points afin de mieux détecter le reste des contours.

## But du seuillage par hystérésis dans le filtre de Canny

Hystérésis est un seuillage à double seuil, comme indiqué avant, si la valeur du pixel  $> \tau_h$  on met le pixel à 255 (blanc, donc contour), contrairement si la valeur du pixel  $w \tau_l$  on met le pixel à 0 (noir, n'est pas un contour). Le plus que va apporter Hysteresis le traitement des points ayant des valeurs comprises entre  $\tau_h$  et  $\tau_b$ , si ce point est connecté à un contour, il est considéré comme contour. Cet algorithme empêche d'avoir des trous dans les contours et évite le bruit local et considéré très efficace.

## Résultats expérimentaux



Figure 2: photographie après le filtre gaussien



Figure 3: Module de gradient de la figure 2



Figure 4: Suppression des non-maximum



Figure 5: photographie après le seuillage par hystéresis  $\tau_l = 33\tau_h = 66$

## Discussion et Conclusion

Il existe quelques imperfections dans les résultats dus aux erreurs de calcul. L'algorithme de Canny reste l'un des meilleurs algorithmes de detection de contours, vu que son taux d'erreurs de signalisation de contour est faible, et il minimise la distance entre les contours detectés et les contours réels