

TP1- MASTER2- ID3D - Septembre -octobre 2019

A. PREAMBULE & prise en main

Ce premier TP (d'analyse d'images) est une prise en main de quelques concepts du premier chapitre. Il est en 2 parties : la première partie de ce travail est d'implémenter les opérateurs de convolution et de les utiliser aussi bien pour lisser le bruit dans une image que pour procéder à une extraction de contours par utilisation d'opérateurs (filtres) différentiels.

Avant de commencer le TP : certains outils sont indispensables pour manipuler les images :
Les TP, seront réalisés de préférence sur OpenCV.

Afin de ne pas passer des heures à traquer des erreurs dont on ignore la réelle origine, votre programme doit avant tout s'exécuter sur des cas simples (voir exemple donné).

Vous pouvez également trouver des exemples sur internet. **Dans tous les cas de figure, vous devez respecter la charte que vous avez signée.**

**** 1ERE PARTIE 1ERE PARTIE 1ERE PARTIE 1ERE PARTIE *****

B. OBJET du TP et implémentation

Ce TP est une application directe du cours. Il permet de voir concrètement :

A.1 Convolution : le filtrage d'image (ou convolution) est un outil fondamental en traitement d'image. OpenCV a de nombreuses fonctions intégrées et efficaces pour effectuer le filtrage d'image, mais vous allez écrire votre propre fonction **convolve_2d** pour cette partie. Il vous est demandé d'écrire un opérateur générique qui traite aussi bien :

1. L'effet d'une convolution d'une image par un filtre de lissage

•	•	•	•	•
•	$h(-1,1)$	$h(0,1)$	$h(1,1)$	•
•	$h(-1,0)$	$h(0,0)$	$h(1,0)$	•
•	$h(-1,-1)$	$h(0,-1)$	$h(1,-1)$	•
•	•	•	•	•

$$\text{Avec } \begin{cases} h(u, v) \geq 0 \\ \sum_{u,v} h(u, v) = 1 \end{cases}$$

2. L'effet d'une convolution d'une image par un filtre différentiel directionnel

$$h(u, v) = \frac{1}{D} H_i(u, v) \quad \text{Où } H_i \text{ est le filtre de direction } i, \quad D \text{ facteur de normalisation}$$

Lissage gaussien : Comme vous l'avez vu dans d'autres cours, il y a plusieurs façons de lisser (ou flouter) une image. Vous pouvez remplacer l'intensité lumineuse du pixel traité par une moyenne ou une moyenne pondérée ou non pondérée calculée sur les pixels voisins. Le flou gaussien est un type particulier de moyennage pondéré des pixels voisins, et est décrit dans les diapositives de la conférence. Pour implémenter le flou gaussien, vous implémenterez une fonction **gaussian_blur_kernel_2d** qui produit un noyau d'une hauteur et d'une largeur données

qui peut ensuite être passé à **convolve_2d** ci-dessus, avec une image, pour produire une version floue de l'image.

$$G_{\sigma} = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

Détection de contours : Dans un premier temps, on procède au calcul du vecteur gradient en chaque point de l'image. La méthode demandée consiste à appliquer des opérateurs (ou masques) de convolution normalisés. A ce niveau différents filtres peuvent être utilisés : Sobel, Prewitt et Kirsch dont les noyaux respectifs sont donnés par les masques horizontaux suivants.

$$H_1 = S_1 = \begin{bmatrix} & & & & \\ & & & & \\ & -1 & 0 & 1 & \\ & -1 & 0 & 1 & \\ & -1 & 0 & 1 & \\ & & & & \end{bmatrix} \quad H_1 = P_1 = \begin{bmatrix} & & & & \\ & & & & \\ & -1 & 0 & 1 & \\ & -2 & 0 & 2 & \\ & -1 & 0 & 1 & \\ & & & & \end{bmatrix} \quad H_1 = K_1 = \begin{bmatrix} & & & & \\ & & & & \\ & -3 & -3 & 5 & \\ & -3 & 0 & 5 & \\ & -3 & -3 & 5 & \\ & & & & \end{bmatrix}$$

3. Une fois l'opérateur de convolution implémenté, passer au processus de la détection de contours : calcul du gradient
 - Calcul de l'amplitude d'un saut (suivant la norme choisie) : attention pour procéder à ce calcul vous devez calculer les composantes suivant les 2 directions canoniques dans le cas d'un détecteur bidirectionnel ou 4 direction dans le cas multi direction (cf. cours)
 - Calcul de la pente du gradient

La détection doit accepter aussi bien un masque bidirectionnel que des masques multi-directionnels (voir explication en séance).

D'autres filtres peuvent éventuellement être proposés et testés (à vous de faire preuve de créativité)

Pour le calcul du gradient : cas discret

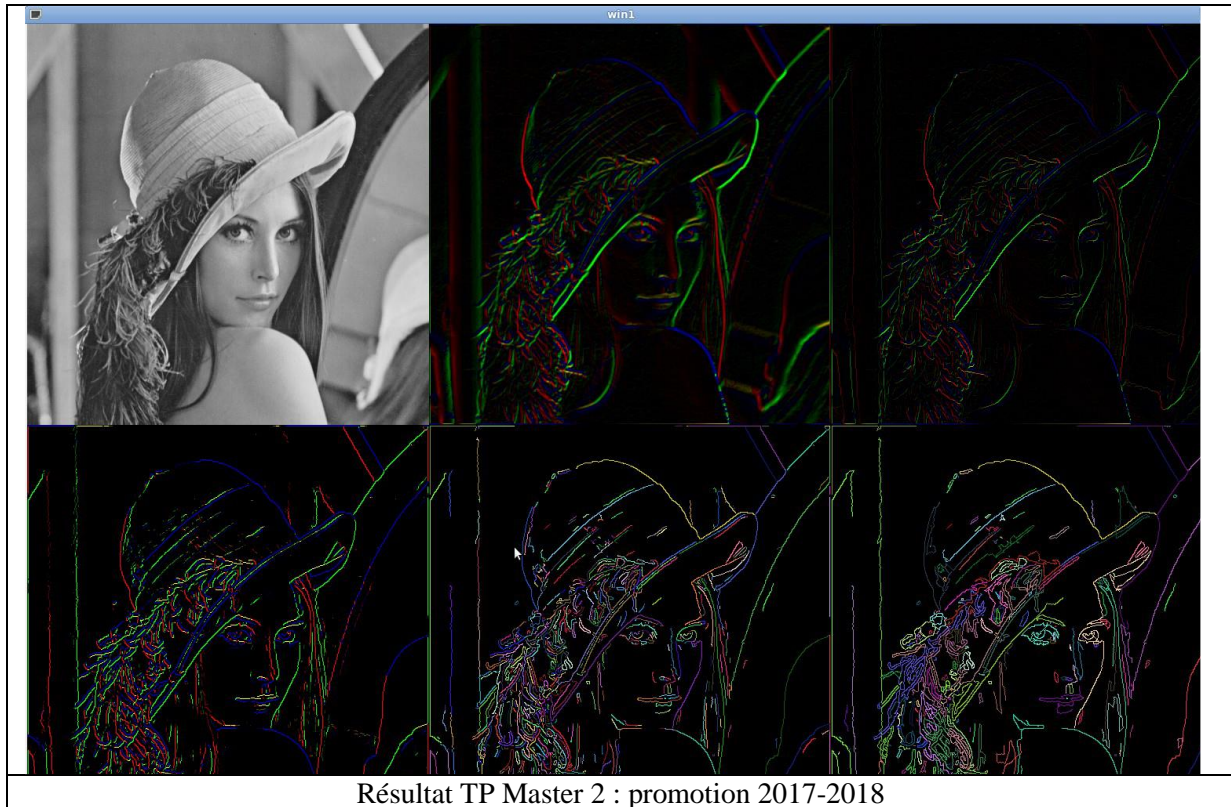
Deux cas sont demandés:

- cas bidirectionnel → pour chaque point on calcule deux composantes qui permettent de donner le MODULE et une PENTE (Cas bi-directionnel)
- cas multidirectionnel → En chaque point on calcule 4 valeurs (voir cours) ces valeurs permettent de déduire un MODULE et une PENTE.

→ Pour ce dernier cas, le calcul du gradient en un point est effectuée par le calcul du produit de convolution avec un masque pour chaque direction potentielle on dispose de 4 directions. On se retrouve avec 4 composantes : en déduire une amplitude et une pente

AFFICHAGE & RESULTAT VISUEL :

Le résultat attendu est d'abord d'ordre visuel. Ainsi l'affichage de l'image résultat (afficher la valeur du gradient - attention aux pièges). La visualisation exigée est une image binaire. En plus de cela, vous pouvez donner d'autres résultat par exemple l'image des directions des contours. Dans ce cas il est possible de jouer avec les couleurs pour donner une idée de ces directions et leurs intensités.



3. Etape du seuillage

Différentes méthodes sont demandées pour effectuer le seuillage d'une image :

A) une méthode à seuil unique :

A.1) une méthode globale : la valeur du seuil est calculée automatiquement en fonction de l'ensemble des valeurs de niveau de gris présentes dans l'image;

A.2) une méthode locale : la valeur du seuil est déterminée localement en fonction d'un nombre restreint de points (dans un voisinage plus ou moins restreint) du point considéré.

B) le seuillage par hystérésis : il s'agit d'un double seuillage du résultat résultant de la détection de contour (voir le cours pour le principe général). Il est demandée de proposer une (ou des) méthode(s) d'estimation des seuils (la valeur ajouté au travail)

➔ Chaque résultat doit être accompagné de commentaires donnant votre avis et votre conclusion

4. Affinage de contours

La détection de contours a pour conséquence de générer, pour chaque zones du contour dans l'image, des points doubles : l'un correspond à la frontière d'une zone, et l'autre à la frontière de la zone

adjacente à la première. L'affinage des contours consiste obtenir des contours d'épaisseur 1. Pour y parvenir on peut utiliser la méthode d'extraction des maxima locaux dans la direction du gradient (par exemple) d'autres stratégies et méthodes peuvent être envisagée (dans ce cas donner la méthode et justifier l'approche).