

## 1. PREAMBULE

Ce premier TP (d'analyse d'images) comporte 2 parties. La première partie concerne la détection de contours par utilisation d'opérateurs différentiels. Ce traitement sera complété par une étape de post-traitement pour affiner les contours obtenus (voir cours).

Avant de commencer le TP : certains outils sont indispensables pour manipuler les images:

- lecture d'image,
- sauvegarde (écriture) d'image
- visualisation d'image

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 (exemples les images jointes à ce fichier)

Vous pouvez également chercher d'autres exemples sur le réseau (internet). **Dans tous les cas de figure, vous devez respecter la charte que vous avez signée**

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

## 2. OBJET du TP

Ce TP est une application directe du cours. Il permet de voir concrètement l'effet d'une détection de contour et comment appliquer les opérateurs présentés dans le chapitre 1 du cours AAI

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 (tableau MxM). A ce niveau différents filtres peuvent être utilisés : Prewitt, Sobel et Kirsch dont les noyaux respectifs sont donnés par les masque horizontaux suivants.

	-1	0	1	
	-1	0	1	
	-1	0	1	

	-1	0	1	
	-2	0	2	
	-1	0	1	

	-3	-3	5	
	-3	0	5	
	-3	-3	5	

D'autres filtres peuvent être définis et appliqués (à vous de faire preuve de créativité)

Au niveau des "spécifications" : la lecture/affichage doit accepter les images de différents formats.

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

### 3. RESULTAT ATTENDU

---

#### IMPLEMENTATION & CALCUL

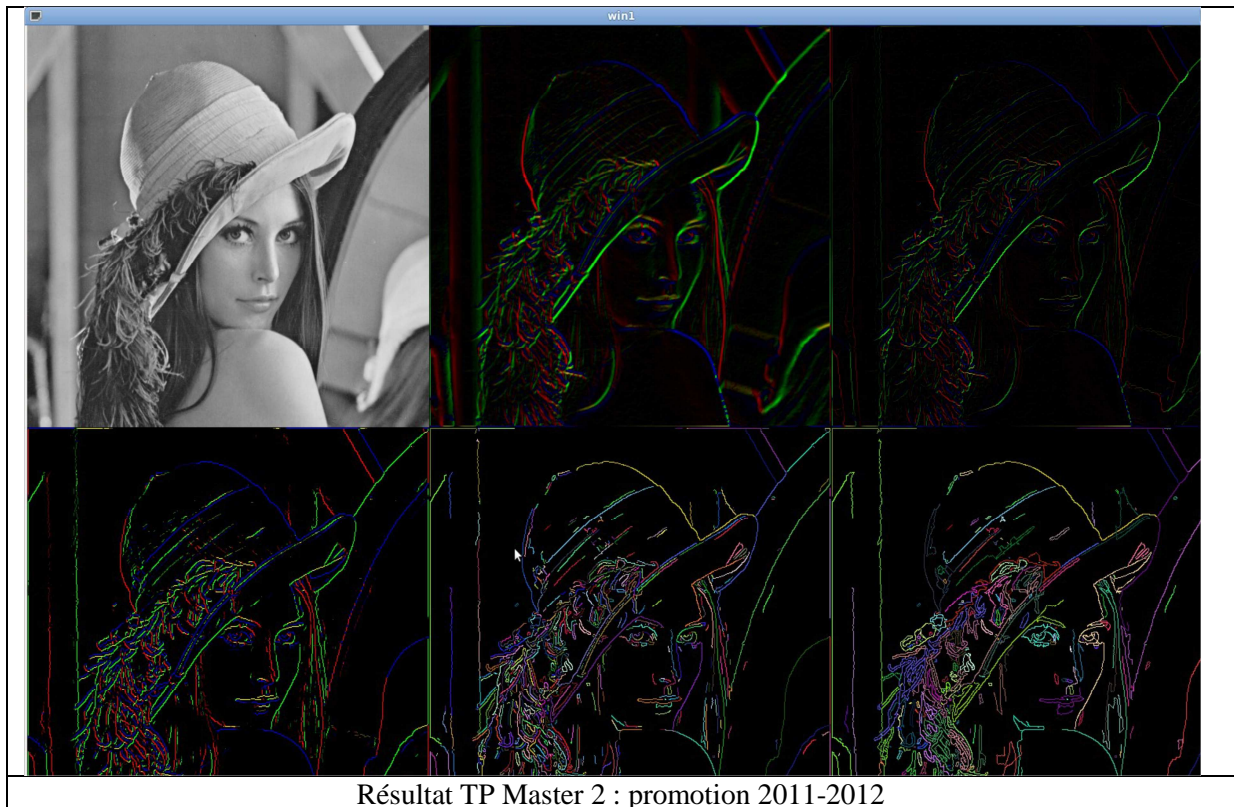
##### 2 cas:

- 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.

→ Ce dernier point n'a pas été souligné en cours, il est présenté en séance de TP. Dans ce dernier cas la détermination de la valeur du gradient en un point est effectuée par le calcul du produit de convolution avec un masque pour chaque direction potentielle → on obtient 4 valeurs.

#### 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). Vous pouvez jouer avec les couleurs pour donner une idée des directions de contours



#### 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).

\*\*\*\*\* 2EME PARTIE 2EME PARTIE \*\*\*\*\*

#### 4. POST-TRAITEMENT

Organisation des contours / extractions des composantes connexes et fermeture de contours (voir le cours + explication donnée en Séance de TP)

##### A. Définition et choix d'implémentation

La formation d'une composante connexe consiste, à partir d'un ensemble de points définissant les contours présents dans une image, à regrouper ces points en des morceaux de courbe permettant d'appréhender un niveau structurel un peu plus élevé que le pixel. Cette opération ne peut évidemment être effectuée qu'après la détection des points contours et affinage (cf. plus haut). Donc cette opération de structuration doit s'effectuer sur une image contour voir quel type d'information utiliser).

##### B. Fermeture de contour

Les traitements appliqués (ci-dessus) aux contours peuvent générer des trous, il est souvent nécessaire de procéder à un "rebouchage" et fermeture de contours. Dans le cours certaines techniques ont été proposées : appliquer ces méthodes ou en proposer une (ou plusieurs) qui permet de fermer les contours.

Un travail de réflexion au choix et l'implémentation des structures de données est nécessaire.

\*\*\*\*\* 3EME PARTIE 3EME PARTIE \*\*\*\*\*

#### 5. Modélisation

Les chaînes obtenues doivent être approximées par une suite de segments de droite ou bien une courbe d'un degré élevé (voir cours de modélisation)

La transformée de Hough est une méthode classique de détection de formes simples dans une image. Cette méthode fait partie de la famille des algorithmes de votes. Ci-dessous se trouvent deux exemples de l'application de la transformée de Hough pour la droite et le cercle.

**Résumé : voir COURS**

La transformée de Hough est un outil utilisé en analyse d'image pour détecter la présence de multiples formes simples ou complexes qui peuvent être exprimées par une forme paramétrique.

Le but : Dans cette partie du TP, vous allez implémenter la transformée de Hough qui permet de réaliser cette détection. Pour des raisons de temps on se limitera dans ce TP à la détection des droites et cercles (ou disque).

I) Principe : on s'appuiera sur le résultat du TP pour la détection et le post-traitement des contours pour obtenir une image binaire

II) Structure de données : la transformée de Hough a besoin d'un choix efficace d'une structure de donnée ,

III) Algorithme général de la transformée de Hough : rappel du principe général, notion de vote, notion de vote pondéré. Quel sont les problèmes qui peuvent apparaître, solution préconisée.

IV) Cas particuliers : avec étude de la complexité

Application à la droite

Application au cercle

---

---

Attention : à tous les niveaux un travail de réflexion au choix et l'implémentation des structures de données est nécessaire.

## CE QUI EST DEMANDE

Un rapport à rendre et une démonstration.

(le délai sera précisé à l'issue de la 1ère séance du TP : en principe au bout de 3 séances)

En plus du rapport, vous devez remettre :

→ Un programme qui tourne et le plus "paramétré" possible

→ Par ailleurs, une interface de manipulation ajoute un bonus au le TP.

→ **Eviter les paramètres mis en dur dans le programme (ça coûte un "malus").**

Le rapport doit être rendu à la date fixée. Le rapport doit mettre en valeur la contribution personnelle du binôme quant à : la structuration des données, le choix des critères, ET la touche personnelle ajoutée.

Pour la **DEMO** : la présence du binôme (les 2) est obligatoire. L'absence d'un membre du binôme lors de la DEMO vaudra un 0 à l'intéressé(e).

Ce qui est attendu au niveau DEMO:

- ✓ Programme compilable
- ✓ "rapide"

- ✓ Visualisation des résultats (affichage) d'images. De façon plus précise, le résultat final doit montrer les performances de votre algo. Ainsi vous aurez à choisir des "best of" ou vous mettez les images les plus favorables pour votre programme. Dans la 2eme partie de la demo afin d'avoir la même mesure pour tout le monde certaines images seront choisies par "l'enseignant".
- ✓

Conseil 1 : avant de tester votre programme sur des images complexes vérifier qu'il marche sur des images simples comme les exemples ci-joints

- ⊕ Les TP sont à réaliser par binômes. La présence aux séances de TP est obligatoire. (la liste des présences est relevée à chaque séance)
- ⊕ Pour les demo, les deux membres du binôme doivent être présents ; la demo doit être faite par les deux.

Pour me donner la possibilité de corriger SURTOUT de tester les TP, il faut joindre par e-mail (obligatoirement) le source. Le rapport doit contenir également un documentation qui permet la compilation du programme. Pour éviter de m'encombrer ma boîte mail, merci de me pas mettre beaucoup d'images.

Attention : les rapports doivent être envoyés par e-mail en précisant les noms et prénoms de chaque participant suivi de son numéro d'étudiant.

Les questions sont à poser en séances de cours ou TP ou bien à envoyer à [saida.bouakaz@univ-lyon1.fr](mailto:saida.bouakaz@univ-lyon1.fr).

Bon courage!!!