

TP: Extraction de contours  
Cours de Traitement d'image  
Master PPMD  
ENSG

Bruno VALLET

IGN/MATIS

## Introduction

Ce TP va vous faire appréhender le problème de l'extraction de contours en vous proposant d'implémenter et d'expérimenter une partie des concepts vus en cours.

## 1 Filtrage dérivatif

1. Trouvez une fonction qui applique un filtre gaussien de variance  $\sigma$  à l'image de votre choix.
2. Trouvez une fonction permettant de réaliser une convolution d'image. En utilisant le cours, construisez des noyaux de convolution simples pour calculer le gradient en colonnes, le gradient en ligne et le Laplacien d'une image.
3. Testez ces opérateurs différentiels sur une image filtrée pour différentes valeurs de  $\sigma$ .
4. Calculez le module du gradient à partir des gradients en ligne et en colonne:

$$\|\mathbf{g}\| = \sqrt{\mathbf{g}_c^2 + \mathbf{g}_l^2}$$

5. Calculez le gradient normalisé:

$$\bar{\mathbf{g}} = \mathbf{g}/\|\mathbf{g}\|$$

6. Calculez le gradient snappé:

$$\mathbf{g}^s = \text{snap}(\bar{\mathbf{g}}/\sin(\pi/8)) \quad \text{snap}(x) = \begin{cases} -1 & \text{si } x < -1 \\ 1 & \text{si } x > 1 \\ 0 & \text{sinon} \end{cases}$$

## 2 Extraction de Contours

1. En utilisant le code de la section précédente et le cours, écrire une fonction `ContoursCanny(image, sigma)` qui renvoie une image binaire des maxima du gradient dans la direction du gradient.
2. Tester cette fonction sur une image. Que constatez vous ?
3. Modifiez la fonction pour qu'elle prenne un seuil en argument, et ne gardez que les maxima dont le module est supérieur au seuil.
4. Comparez avec un seuillage sur le module du gradient. Quel est l'intérêt de cette méthode ?
5. Seuillez le Laplacien d'une image (à 0) pour différents  $\sigma$ .
6. Ecrire une fonction `ContoursLaplace(image, sigma)` qui renvoie une image binaire des pixels de Lapacien positif qui ont au moins un voisin de Laplacien négatif.
7. Modifiez la fonction pour qu'elle prenne un seuil en argument, et ne gardez que les pixels pour lesquels le max des différences (en niveau de gris) avec un de ses voisins négatifs est supérieure à un seuil donné.
8. Quelles sont les différences notables avec l'approche précédente ?

## 3 Optionel: Filtrage par hystérésis

1. Implémenter une fonction `Cluster(contours)` qui regroupe les pixels adjacent d'un masque de contours (avec un seuil bas) dans une liste de listes de paires  $(l, c)$ . On pourra utiliser une pile pour stocker les pixels à traiter et une image binaire indiquant si un pixel a déjà été traité.
2. Implémenter une fonction `Hyst(contours, gradnorm, seuil)` qui produit un nouveau masque de contours avec seulement les clusters ayant au moins un pixel pour lequel le module du gradient est supérieur au seuil.
3. Remplacer le seuil dans les fonctions `ContoursCanny` et `ContoursLaplace` par deux seuils hauts et bas, et utiliser la fonction `Hyst` pour appliquer le seuillage par hystérésis aux résultats de ces deux fonctions. Pour `ContoursLaplace`, on pourra stocker la différence de niveaux de gris sur laquelle agit le seuil pour la passer à `Hyst`