

TP3 - Hiver 2018

IMN 259

Analyse d'images

Date limite pour remettre votre travail : 23 mars

Objectifs

Implémenter une opération ponctuelle et extraire des caractéristiques

1. Détection de contours
 - (a) Seuil de la norme du gradient
 - (b) Suppression des non-max
 - (c) Zero-Crossing
2. Détection de coins
3. Segmentation
4. Sauvegarde d'images de transformée de Fourier.

Description

À l'aide du code C++ fourni (fichiers tp3A.cpp, tp3B.cpp, tp3C.cpp, tp3D.cpp, tp3E.cpp, tp3F.cpp, MImage.h, MImage.cpp, matrixrgb.h) vous devez implémenter différentes fonctions vues dans le cours (une par fichier tp3X.cpp). Pour ce faire, il est fortement recommandé de récupérer les fonctions *EdgeDetec*, *NonMaxSupp*, *HarrisCornerDetec*, *ZeroCrossing*, *SaveSpectralImage*, *CyclRecal* et *KMeansSegmentation* de la classe *MImage*. Toutefois, vous êtes libre d'ajouter d'autres fonctions à la classe *MImage* si vous en éprouvez le besoin. Évidemment, il est fortement recommandé de récupérer du code des travaux pratiques 1 et 2.

Recommandations :

1. tp3A : **Seuil de la norme du gradient**. Pour ce numéro, vous devez implémenter l'algorithme B1 du «seuil du gradient» présenté dans le chapitre sur l'extraction de caractéristiques.
2. tp3B : **Zero-Crossing**. Pour ce numéro, vous devez implémenter l'algorithme B7 du «zero-crossing» présenté au chapitre sur l'extraction de caractéristiques. Toutefois, il vous faut ici remplacer les 0 par le seuil passé en paramètre (variable *threshold*). Vous devez également implémenter le filtre laplacien *LaplacianFilter* ainsi qu'un filtre passe bas (gaussien ou moyennneur)
3. tp3C : * **Suppression des non-max**. Vous devez implémenter l'algorithme B4 de *thinning* présenté au chapitre sur l'extraction de caractéristiques. Vous n'avez pas à appliquer de filtre passe bas (gaussien ou autre).
4. tp3D : * **Détecteur de Harris**. Vous devez implémenter l'algorithme suivant :
 - (a) Calculer la dérivée en X de l'image d'entrée $f(x, y) : f_x = \partial f / \partial x$.
 - (b) Calculer la dérivée en Y de l'image d'entrée $f(x, y) : f_y = \partial f / \partial y$.

(c) Pour chaque pixel (x, y) , calculer les valeurs A, B et C de la matrice hessienne

$$\begin{aligned} A &= \sum_{i,j} W_{i,j} f_x(i, j)^2 \\ B &= \sum_{i,j} W_{i,j} f_x(i, j) f_y(i, j) \\ C &= \sum_{i,j} W_{i,j} f_y(i, j)^2 \end{aligned}$$

où $W_{i,j}$ est un coefficient gaussien (utilisez la fonction `MGaussian2D`).

(d) Calculer $M_c(x, y) = \det(H) - k \cdot \text{trace}(H)^2$.

(e) Lorsque tous les pixels auront été visités, ramener la dynamique de l'image M_c entre 0 et 255 (fonction `Rescale`).

(f) Sauvegarder M_c

5. tp3E : **K-moyennes**. Vous devez implémenter l'algorithme B13 des K-moyennes présenté au chapitre sur l'extraction de caractéristiques.
6. tp3F : **Recalage logarithmique et cyclique d'une TF**. Pour ce travail, les fonctions effectuant la TF ainsi que la TF inverse vous sont fournis (fonctions `FFT` et `IFFT`). Vous devez toutefois implémenter la fonction "SaveSpectralImage" qui a pour objet de sauvegarder la magnitude d'une TF. Pour ce faire, vous devez calculer la magnitude du signal à chaque pixel (i, j) de la TF :

$$\frac{\sqrt{\text{Re}(i, j)^2 + \text{Im}(i, j)^2}}{NM} \quad (1)$$

où N et M contiennent la taille de l'image. À noter que les parties réelle et imaginaire sont comprises dans les canaux `r` et `g`. En d'autres mots : $\text{Re}(i, j) \rightarrow \text{GetColor}(i, j, 0)$ et $\text{Im}(i, j) \rightarrow \text{GetColor}(i, j, 1)$. Lorsque le paramètre "logTransform == true", vous devez effectuer un recalage logarithmique, *i.e.*

$$250 * \log\left(\frac{\sqrt{\text{Re}(i, j)^2 + \text{Im}(i, j)^2}}{NM} + 1\right). \quad (2)$$

Faites attention à ce que cette opération ne retourne pas de valeurs supérieures à 255. Évidemment, il est **FORTEMENT** recommandé d'utiliser la fonction "SaveImage" que vous avez créé lors du TP1.

Vous devez également implémenter un recalage cyclique. Pour ce faire, il vous faut compléter la fonction "CyclRecal".

Afin de vous simplifier la vie, vous n'avez qu'à traiter des **images en niveau de gris** pour ce tp.

(*) Problèmes un peu plus difficiles que je vous recommande de faire en dernier.

Évaluation

Ce travail doit être fait en **équipe de DEUX**. Pour simplifier la correction, nous vous demandons de modifier le moins possible les fonctions **main**. Veuillez noter que le barème d'évaluation est disponible avec le reste du code. Au moment de soumettre votre travail, assurez-

vous que votre code compile bien sous Linux (vous n'avez qu'à taper la commande *make* dans un terminal Linux). Utilisez le système **turnin web** : <http://opus.dinf.usherbrooke.ca:8080/> pour soumettre **TOUS LES FICHIERS NÉCESSAIRES POUR COMPILER VOTRE TRAVAIL** ainsi que toutes les images.