

TP2 - Hiver 2018

IMN 259

Analyse d'images

Date limite pour remettre votre travail : 23 février

Objectifs

Implémenter les opérations ponctuelles que voici :

1. Correction gamma
2. *Warping*
 - (a) Interpolation au plus proche voisin
 - (b) Interpolation bilinéaire
3. Filtrage médian temporel
4. Égalisation d'histogramme

Description

À l'aide du code C++ fourni (fichiers tp2A.cpp, tp2B.cpp, tp2C.cpp, tp2D.cpp, MImage.h, MImage.cpp, matrixrgb.h) vous devez implémenter différentes opérations vues dans le cours (une par fichier tp2X.cpp). Pour ce faire, il est fortement recommandé de récupérer les fonctions *HistogramEqualization*, *WaveWarping*, *TemporalMedianFilter* et *GammaCorrect* de la classe *MImage*. Toutefois, vous être libre d'ajouter d'autres fonctions à la classe *MImage* si vous en éprouvez le besoin. Pour lire et sauvegarder les images, vous devez récupérer le code produit dans le cadre du tp1. SVP ne pas changer le contenu des fonctions **main**.

Recommandations pour ce travail :

1. tp2A : **Égalisation d'histogramme**. Pour ce numéro, il vous faut égaliser l'histogramme d'une image en niveaux de gris. Pour ce faire, il est fortement recommandé de suivre les étapes suivantes :
 - (a) Calculer l'histogramme normalisé $P(c)$ de l'image d'entrée $f(x, y)$.
 - (b) Partant de $P(c)$, calculer la fonction cumulative $T(r)$

$$T(r) = \sum_{c=0}^r P(c).$$

- (c) Égaliser $f(x, y)$ à l'aide de la fonction de transfert $T(r)$.
2. tp2B : **Déformation géométrique de l'image**. Vous devez implémenter une fonction vous permettant de «warper» une image d'entrée. En vertu de la notation introduite dans les notes de cours, vous devez implémenter la déformation suivante :

$$\begin{pmatrix} i \\ j \end{pmatrix} = \begin{pmatrix} i' + 10 \sin(2\pi \frac{j'}{100}) \\ j' + 10 \sin(2\pi \frac{i'}{100}) \end{pmatrix} \quad (1)$$

où (i, j) est la coordonnée euclidienne d'un pixel dans l'image d'entrée et (i', j') est la coordonnée euclidienne d'un pixel dans l'image déformée. Ceci dit, vous pouvez également

implémenter une transformation géométrique de votre cru. La seule contrainte est que le résultat final doit être cohérent.

À noter qu'il vous faudra implémenter une interpolation par le **plus proche voisin** et une interpolation **bilinéaire**.

3. tp2C : **Correction gamma**. Vous devez implémenter une correction gamma sur tous les pixels de l'image d'entrée, *i.e.*

$$g(x, y) = (f(x, y))^\gamma \quad (2)$$

où $g(x, y)$ est l'image de sortie. À noter que pour cette dernière équation, les niveaux de gris de l'image $f(x, y)$ doivent être normalisés entre 0 et 1.

4. tp2D : **Filtrage médian temporel**

Ce programme a pour objet de détecter la présence ou l'absence de mouvement dans une séquence vidéo. Mais pour ce faire, il faut au préalable estimer l'image du fond (l'image $B(x, y)$ dans les notes de cours). Pour ce faire, il vous faut appliquer un filtre médian temporel sur les 50 images du répertoire «videoImages».

Sauf pour l'égalisation d'histogramme, les fonctions doivent être en mesure de traiter des images couleurs et des images en niveaux de gris. Pour les images couleurs, vous devez traiter chacune des bandes individuellement.

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** et de remettre TOUS les fichiers incluant le Makefile, les images et les fichiers tpYY.cpp. Veuillez noter que le barème d'évaluation est disponible avec les fichiers .h et .cpp. 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 turnin web (<http://opus.dinf.usherbrooke.ca:8080/>) pour soumettre votre travail. À noter qu'une pénalité de 10% par jour sera appliquée à tout travail remis en retard.