

TP2 - Hiver 2018

IMN 259

Analyse d'images

Date limite pour remettre votre travail : 15 février à 23h59

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

Vous devez coder vos propres fonctions. Pas d'appels à des fonctions existentes en python.
Le but est d'apprendre à coder ces opérations simples et jouer avec des images en python.

Description

À l'aide du code Python fourni (fichiers *tp2.py*, *MI_image.py*) vous devez implémenter différentes opérations vues dans le cours. Pour ce faire, il est fortement recommandé de récupérer les fonctions *HistogramEqualization*, *WaveWarping*, *TemporalMedianFilter* et *GammaCorrect* du fichier *MI_image*. Toutefois, vous êtes libre d'ajouter d'autres fonctions à *MI_image* si vous en éprouvez le besoin. Pour lire et sauvegarder les images, vous devez récupérer le code produit dans le cadre du tp1.

Recommandations pour ce travail :

1. **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. **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 choix. 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. **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. A noter que pour cette dernière équation, les niveaux de gris de l'image $f(x, y)$ doivent être normalisés entre 0 et 1.

4. **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 OU TROIS**. Remettez un rapport simple rapportant dans une table toutes les mesures et images sorties des opérations ponctuelles pour chacune des images données. Au moment de soumettre votre travail, assurez-vous que votre code roule bien et que tous les fichiers nécessaires sont soumis. Si vous installez des librairies, svp fournir le fichier requirements.txt.

IMPORTANT

Si vous utilisez l'IA générative ou autres sources (*Google, StackOverflow, etc.*) pour vos solutions, vous devez les citer dans votre code. Je me réserve de vous demander à l'oral de m'expliquer le contenu de vos codes à tout moment après la remise.