

# TP2 - Hiver 2018

IMN 259

## Analyse d'images

Date limite pour remettre votre travail : 18 février à 23h59

## Objectifs

Implémenter les opérations et traitements d'images que voici :

1. *Warping*
  - (a) Interpolation au plus proche voisin
  - (b) Interpolation bilinéaire
2. Bruit et rapport signal sur bruit
3. Filtrage médian temporel

## Description

À l'aide du code Python fourni (fichiers *tp2.py*, *MI\_image.py*, *snr.py*) vous devez implémenter différentes opérations vues dans le cours. Pour ce faire, il est fortement recommandé de mettre votre code directement dans les fonctions *WaveWarping*, et *TemporalMedianFilter* 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.

1. **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**. Comme dans mes diapos, montrez le résultat des deux méthodes d'interpolation. Décrire brièvement ce que vous observez sur les images données.

2. **Bruit et rapport signal sur bruit (SNR).** Sur l'image du carré blanc sur fond noir, bruitez l'image avec 3 types de bruit : i) bruit uniforme (*np.random.uniform*), ii) bruit Gaussien (*np.random.normal*), et iii) bruit poivre et sel (comme dans les diapos).
  - (a) Illustrez l'image originale et sa version bruitée.
  - (b) Illustrez son histogramme (avec matplotlib ou votre code du tp1).

- (c) Calculez le SNR théorique avec mon code de la métrique (*snr.py*) et avec l'approche numérique approximative (moyenne / écart-type-du-bruit) (dém03).
- (d) Démontrez que la moyenne de  $N$  images bruitées avec un bruit indépendant équivaut avec une meilleure image proportionnelle à un SNR de  $\sqrt{N}$ . Est-ce que votre réponse change en fonction du type de bruit que vous ajoutez ?

Décrivez brièvement vos observations et résultats.

3. **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». Expliquez votre solution et vos résultats.

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 UN ou DEUX**. 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.