IMN530 - RECONSTRUCTION ET ANALYSE D'IMAGE MÉDICALE

TP 2 - Recalage

- Valeur du TP : 20%
- Total : sur 150
- Vous devez remettre un rapport PDF. Le style et la forme de vos réponses sont libres.
- Vous pouvez travailler en équipe de 2-3 ou seul. Soyez originaux.
- La date de remise du TP sera déterminée en classe.
- Pour chaque étape, décrivez votre démarche, sauvegardez vos commandes et décrivez le code utilisé. Le rapport est aussi important, sinon plus, que le code. Prenez le temps de décrire ce que vous avez fait et pourquoi.
- Codez en python. Trucs pour faire du « beau » code (ce sera vérifié!)
    - Votre script (main) devrait contenir très peu de code, et surtout appeler les sous-fonctions que vous aurez créées. Utilisez un argparser. Le nom des fichers (ex, path des images) devrait être envoyé en paramètre, et non être écrit directement dans le code.
    - Python est très lent pour les boucles for imbriquées. Tentez de remplacer les boucles par des calculs sur les matrices.

Question 1. [20 pts] Histogramme conjoint

a) Créer un script `show_joint_hist.py` qui devra appeler une fonction `joint_hist(I, J, bin)` qui calcule l'histogramme conjoint de deux images (de même taille) I et J en divisant leur intervalle de valeurs en bin sous-intervalles.

*** Attention à ne pas garder les images dans leur type original si elles sont loadées par défaut en utin8! Les convertir en int ou en float avant!*
*** L'échelle logarithmique peut être un bon choix pour la visualisation.*
*** Vous devez coder vous-mêmes votre histogramme, et non appeler une librairie*

b) Pour des images de taille n x p, vérifiez dans votre code que :

$$\sum_{i,j} H_{I,J}(i,j) = n * p$$

c) Calculez et affichez l'histogramme conjoint des différents couples (Ik, Jk) fournis avec le TP. Décrivez ce que vous voyez pour chaque paire.

Question 2. [30 pts] Critère de similarité

a) Créer un script `compare_pairs.py` qui devra appeler, pour 2 images I et J de même taille,

- une fonction `ssd(I, J)` qui calcule la somme des différences au carré
- une fonction `cr(I, J)` qui calcule le coefficient de corrélation
- une fonction `IM(I, J)` qui calcule l'information mutuelle.

b) Comparez les résultats de ces trois fonctions sur les différents couples d'images (I_k, J_k) fournis. Décrivez ce que vous remarquez.

*** Codez vous-mêmes vos fonctions SSD et CR. Elles peuvent être calculées à partir des images ou à partir des histogrammes. Vous pouvez appeler des librairies pour la fonction IM.*

Question 3. [30 pts] Transformations spatiales

a) Générez une grille régulière de points 3D (voir Figure 1a)

b) Créer un script `transform_grid.py` qui devra appeler l'une ou l'autre des fonctions ci-dessous.

i. Écrivez une fonction `trans_rigide(theta, omega, phi, p, q, r)` qui renvoie la matrice (en coordonnées homogènes) de la transformation rigide correspondant à faire:

i. une rotation d'angle theta autour de l'axe x
ii. une rotation d'angle omega autour de l'axe y
iii. une rotation d'angle phi autour de l'axe z
iv. une translation du vecteur t = (p, q, r)

Testez cette fonction sur le nuage de points 3D réguliers de a) et affichez le résultat (exemple sur la Figure 1b).

ii. Écrivez une fonction `similitude` qui ajoute une homothétie (scaling) de rapport s. Affichez et testez cette fonction comme en ii) (exemple en figure 1c).

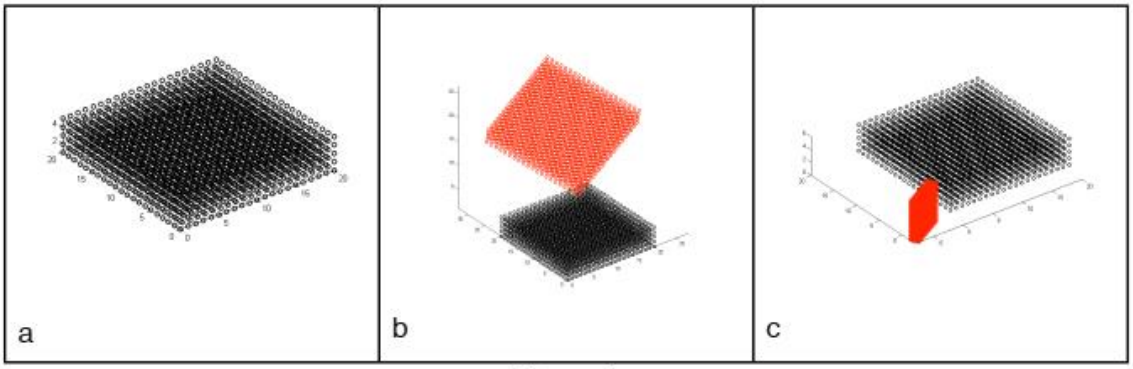

Figure 1

c) Étant donné les 3 matrices suivantes: M1, M2, M3. Déterminer le type de transformations correspondantes. Justifiez.

M1 =
0.9045	-0.3847	-0.1840	10.0000
0.2939	0.8750	-0.3847	10.0000
0.3090	0.2939	0.9045	10.0000
0	0	0	1.0000

M2 =
-0.0000	-0.2598	0.1500	-3.0000
0.0000	-0.1500	-0.2598	1.5000
0.3000	0	0	0
0	0	0	1.0000

M3 =
0.7182	-1.3727	-0.5660	1.8115
-1.9236	-4.6556	-2.5512	0.2873
-0.6426	-1.7985	-1.6285	0.7404
0	0	0	1.000

Question 4. [50pts] Recalage iconique 2D simple

Créer un script `register_image.py` qui appellera une ou plusieurs des fonctions ci-dessous.

a) Écrivez une fonction `translate_image(I, p, q)` qui retourne une nouvelle image correspondant à l'image I translatée du vecteur t = (p, q). Les composantes p et q peuvent être des float. Vous devez donc gérer l'interpolation. Appelez des fonctions d'interpolation existantes.

b) Écrivez une fonction `register_translation_ssd(I, J, ...)` qui performe un recalage 2D vu en cours en minimisant la SSD et en considérant uniquement des translations. Ajoutez vos paramètres à l'appel de cette fonction et décrivez vos paramètres dans le rapport.

i. Au cours de l'évolution sauvegardez « l'énergie » SSD correspondant à chaque état.

ii. Suggestion. Au cours de l'évolution, affichez l'image modifiée après chaque itération. Cela vous aidera à déboguer le code.

c) Testez votre fonction pour 3 translations différentes de l'image de cerveau Brain_MRI_1 (que vous aurez créées vous-mêmes! Pas Brain_MRI2,3,4!).

i. Visualisez le recalage obtenu.

ii. Décrivez quantitativement la qualité du recalage obtenu.

iii. Visualisez la courbe de d'énergie et décrivez-la. Est-elle strictement décroissante? Si elle n'est pas strictement décroissante, pourquoi, à votre avis? Que faudrait-il faire pour être plus correct?

d) Écrivez une fonction `rotate_image(I, theta)` qui produit une image correspondant à l'image I à laquelle on a appliqué une rotation d'angle theta et de centre (0,0) qui est au coin en haut à gauche de l'image.

e) Écrivez une fonction `register_rotation_ssd(I, J, ...)` qui performe un recalage 2D en minimisant la SSD et en considérant uniquement des rotations.
   i. Répondez aux mêmes question qu'en c.

f) Écrivez une fonction `register_rigid_ssd(I, J, ...)` qui performe une descente du gradient pour minimiser la SSD en considérant cette fois les transformations rigides (translations + rotations).
   i. Recalez BrainMRI_2,3,4 sur Brain_MRI_1.
   ii. Quels recalages convergent? Lesquels ne convergent pas? Que se passe-t-il à votre avis pour ceux qui ne convergent pas?
   iii. Quantifiez la qualité du résultat.

g) Pour améliorer les performances de la descente du gradient à pas fixe, il faut une technique d'optimisation un peu plus évoluée. Améliorez votre recalage rigide avec une meilleure technique d'optimisation de votre choix.
   i. Testez pour 3 cas de transformations rigides variés.

Question 5. [20 pts] Préparation pour le TP3/projet.

Installer le logiciel ANTS (http://picsl.upenn.edu/software/ants/).

i. Regardez ce qui existe pour faire du recalage linéaire et recalage non-linéaire. Décrivez les paramètres.
ii. Prenez la flair.nii et t1.nii du TP1 et rouler ANTs linéaire et nonlinéaire.
iii. Quels sont les meilleurs paramètres à votre avis (critère de similarité, transformation et interpolation)? Le non-linéaire est-il meilleur?

*** Ne sous-estimez pas cette dernière question si vous n'avez pas encore installé ANTS! Ça peut être long à installer et long à rouler!*