

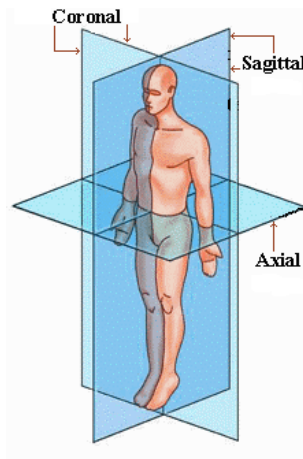
# TP3

## Segmentation d'une image volumique par l'algorithme des k-moyennes

pierre.maurel@irisa.fr

Vous trouverez, dans le fichier `IRM_cerveau.mat`, une matrice `M0` de taille  $143 \times 198 \times 154$  correspondant à une image volumique. Cette image volumique a été obtenue à partir d'une IRM de la tête : après une opération de masquage, seul le cerveau a été gardé et les pixels du fond ont été fixés à `NaN`.

**Exercice 1** Écrivez une fonction permettant d'afficher, en les faisant défiler, les différentes coupes du volume. Cette fonction prendra en paramètre une option indiquant quelle vue (axiale, sagittale, coronale) est choisie.



Vous pourrez utiliser la fonction `squeeze` qui prend une matrice de dimension  $n \times 1 \times p$  ou  $1 \times n \times p$  et renvoie la matrice de dimension  $n \times p$  correspondante.

**Exercice 2** Programmez une fonction `Seg = k_moyennes(I,k)` qui segmente une image `I` en `k` classes en utilisant l'algorithme des k-means. Ne prenez pas en compte les pixels correspondant au fond (`NaN`) qui constitueront une classe à part entière.

### Remarques

- Un squelette de cette fonction est fourni.
- Pour des images en niveau de gris, l'algorithme des k-means s'effectue en 1D. Cela revient donc à choisir des seuils séparant les classes et à mettre à jour ces seuils jusqu'à que leur valeur ne change plus.

**Exercice 3** Testez votre fonction pour segmenter automatiquement l'IRM 3D du cerveau en 4 classes : le fond, le liquide céphalo-rachidien, la matière grise, la matière blanche. La segmentation se fait sur le volume 3D mais vous pourrez afficher, par exemple, la coupe  $M0(:, :, 80)$  ainsi que la segmentation correspondante. Pour afficher en fausse couleur : `imagesc(I);axis equal; axis off;`

**Exercice 4** La fonction `MarchingCubes` fournie est équivalente à la fonction `isosurface` (regardez `help isosurface`) de matlab mais est implémentée de manière plus efficace. En vous en servant, affichez la surface corticale (extérieur de la matière grise, obtenue par votre segmentation). Vous pourrez utiliser la fonction `smooth3` (sur le résultat de votre segmentation) pour obtenir une surface plus lisse.

**Exercice 5** La première image fournie correspondait bien à une acquisition IRM réelle mais elle avait été choisie pour sa qualité.

Le fichier `IRM_cerveau_avecbiais.mat` contient également une matrice  $M0$  correspondant à une image volumique d'IRM de la tête pour laquelle on a extrait le cerveau. Cette image présente un défaut très courant en IRM : un biais dû à une inhomogénéité du champ magnétique  $\vec{B}_0$  lors de l'acquisition. Testez votre algorithme de segmentation sur cette image, que se passe-t-il ?

**Exercice 6** Pour mettre en évidence ce biais, représentez sous forme de surface une coupe de l'image : `surf(M0(:, :, 80))`. Que constatez-vous ? Que proposeriez-vous pour corriger ce biais et améliorer la segmentation par l'algorithme des k-moyennes ?

**Exercice 7** Le fichier `plan_least_square.m` contient un exemple de régression par moindres carrés : étant donné un nuage de points  $(x_i, y_i, z_i)$  on cherche le plan, d'équation  $z = ax + by + c$ , qui approxime le mieux le nuage, i.e. qui minimise la somme des distances au carré entre chaque point et le plan. Inspirez vous de ce fichier pour corriger le biais dans la coupe  $M0(:, :, 80)$  et comparez alors le résultat de la segmentation obtenu avant et après correction.

Vous pourrez commencer par tester votre fonction de correction de biais sur `image_test_biais.png`.

