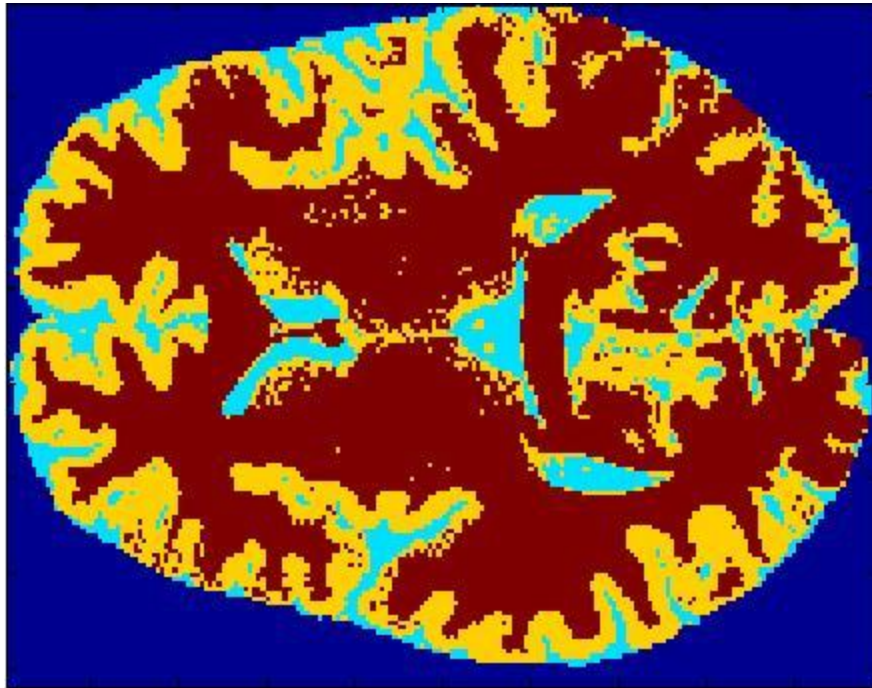


Compte Rendu TP1

IMED1



Thomas Davenel - Maël Le Vot

18/11/2021

ESIR 3

Introduction

L'objectif de ce TP est de traiter des images issues d'un volume 3D représentant un cerveau humain et d'en identifier les zones. Le volume 3D en question a été construit à partir d'une IRM de la tête et d'une opération de masquage utilisée pour effacer les parties de la tête non utiles pour les opérations à venir. Ces dernières consistent à afficher les coupes du volume, à les segmenter afin d'en identifier les régions mais aussi de traiter et de corriger des coupes issues d'une IRM défectueuse. Une représentation 3D de certaines surfaces liées au cerveau sera également donnée.

Affichage des coupes du volume

Nous devons ici écrire une fonction permettant d'afficher les différentes coupes du volume 3D du cerveau à notre disposition (appelé `M0` dans la suite du rapport). De plus, il nous faudra discerner les vues coronale (axe `y`) , sagittale (axe `x`) et axiale (axe `z`) selon l'orientation de `M0` que nous voulons visualiser.

En définitive, nous arrivons à représenter l'évolution des coupes du cerveau selon l'orientation choisie dont voici quelques exemples :

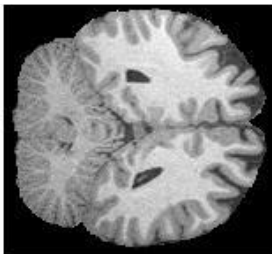


Figure 1 : Coupe coronale
`M0(:,133,:)`



Figure 2 : Coupe sagittale
`M0(103,:,:)`

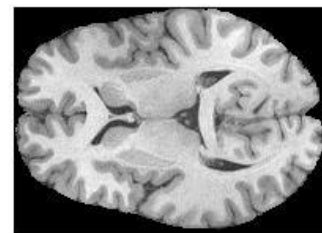


Figure 3 : Coupe axiale
`M0(:, :,80)`

Maintenant que nous sommes en capacité de récupérer n'importe quelle coupe de `M0`, nous pouvons réfléchir à la manière de les segmenter afin d'en identifier les zones importantes.

Segmentation des coupes de l'IRM

Afin de segmenter les différentes parties du cerveau sur les images mises à disposition, nous allons utiliser la technique de segmentation des k-means.

On rappelle tout d'abord que le principe de cet algorithme est de rechercher la meilleure répartition des pixels (ici en niveaux de gris) en k classes.

En niveau de gris, la segmentation par la méthode des k-means se fait dans une dimension (le niveau de gris entre 0 et 255). Ainsi, effectuer la méthode des k-means revient à trouver k+1 seuils (en comptant la valeur minimale et la valeur maximale comme seuil) pour diviser l'espace de niveau de gris en k parties, et mettre à jour ces seuils jusqu'à ce qu'ils ne changent plus.

On peut donc maintenant tester cet algorithme sur une coupe axiale, $M0(:,:,80)$:

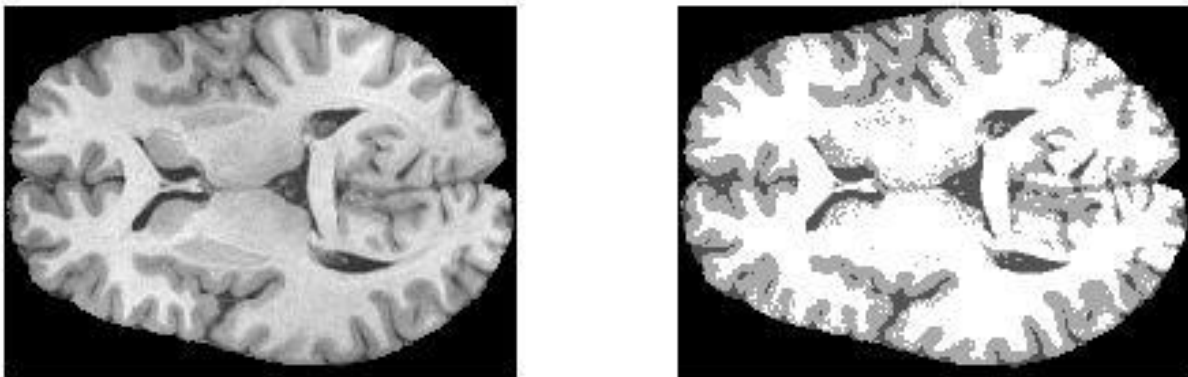


Figure 4 : Comparaison entre la coupe axiale $M0(:,:,80)$ (à gauche) et sa segmentation en 4 classes (à droite)

Afin de mieux visualiser les différentes parties du cerveau, nous allons afficher la segmentation précédente (Figure 4) en fausse couleur grâce à la fonction *imagesc*:

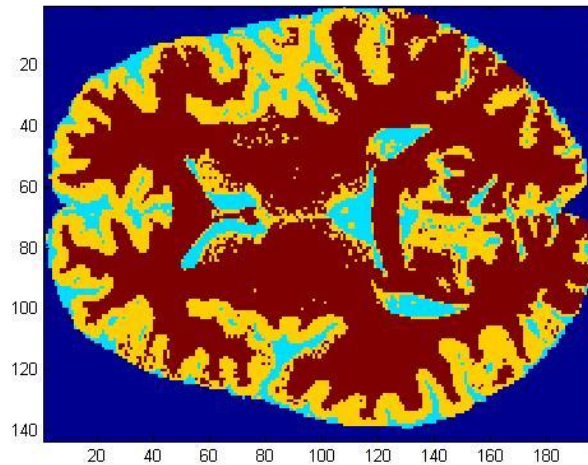


Figure 5 : Affichage en fausses couleurs de la segmentation de la coupe axiale $M0(:, :, 80)$

Sur la figure 5, on peut distinguer les différentes composantes d'un cerveau humain :

- Le bleu correspond au fond de l'image et ne fait pas partie du cerveau.
- Le bleu ciel est le liquide céphalo-rachidien, il permet d'amortir les mouvements ou les chocs qui risqueraient d'endommager le cerveau.
- Le jaune correspond à la matière grise du cerveau, elle comprend les neurones.
- La partie marron correspond à la matière blanche du cerveau.

Dans un cas d'application réel, cette représentation peut permettre à un chirurgien d'identifier non seulement les zones du cerveau du patient, mais aussi d'éventuelles lésions cérébrales.

Représentation 3D

Maintenant que nous savons segmenter n'importe quelle coupe du cerveau, nous pouvons en donner la représentation 3D en s'appuyant sur un équivalent plus performant de la fonction Matlab *isosurface* nommé *MarchingCubes* (fonction fournie). Cette fonction permet de représenter une isosurface, c'est-à-dire une représentation en 3 dimensions d'une distribution de données constantes également en 3 dimensions. Dans les faits, nous construisons un volume segmenté en 3 classes sur la vue axiale à partir du volume $M0$ et de la fonction k-means appliquée à un volume. Ensuite, nous lisons la surface obtenue grâce à la fonction *smooth3* de Matlab et nous lui appliquons la fonction *MarchingCubes*.

Cependant, nous souhaitons afficher la surface corticale du cerveau, c'est-à-dire la surface située entre le liquide céphalo-rachidien et la matière grise. Par conséquent, pour savoir quel paramètre placer dans la fonction, il nous faut déterminer comment les parties segmentées de la coupe choisie (ici, M0(:, :, 80)) ont été indexées par Matlab. Ces vérifications se trouvent sur les figures 6 et 7.

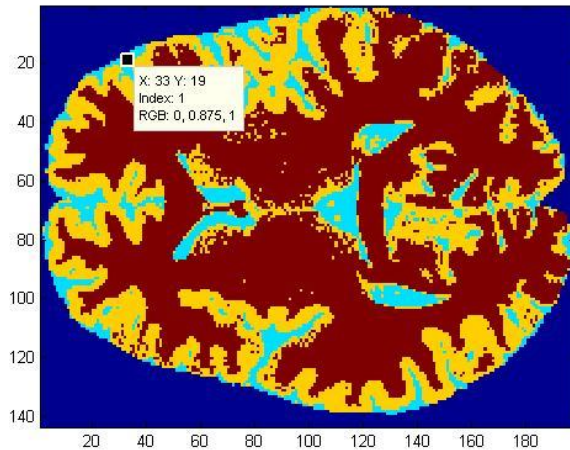


Figure 6 : Index du liquide céphalo-rachidien

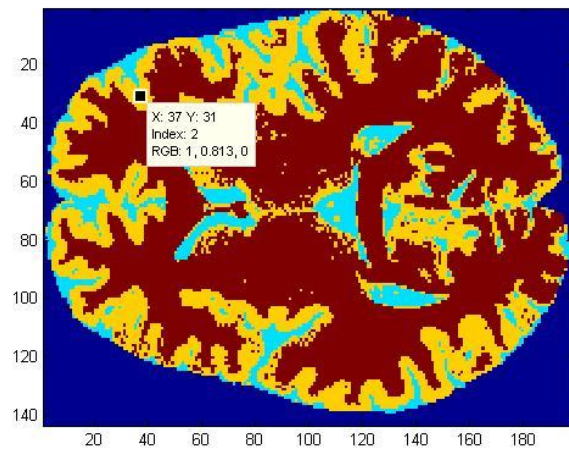


Figure 7 : Index de la matière grise

Comme nous pouvons le constater, l'index du liquide est le **1** et celui de la matière grise est le **2**. Pour obtenir la surface corticale, il nous suffit donc de donner la valeur de $(1+2)/2$, soit 1.5 au paramètre de la fonction. On peut alors afficher la surface sur la figure 8.

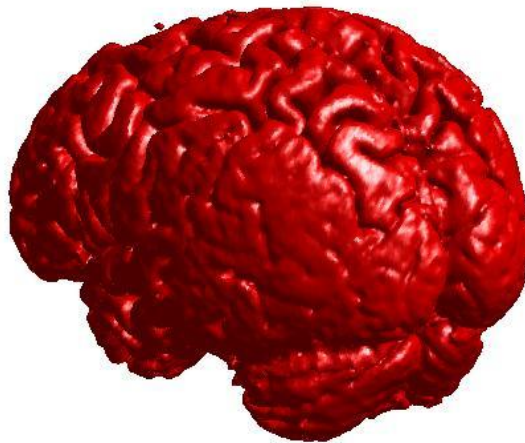


Figure 8 : Surface corticale du cerveau à la coupe M0(:, :, 80)

Cette représentation peut être utile lors de l'observation d'un patient par un médecin pour prévenir d'éventuels risques cérébraux, voire même pour situer précisément une tumeur. Dans la suite, nous ferons face à un volume issu d'un IRM défectueux qui nous demandera d'effectuer des opérations supplémentaires pour pouvoir le traiter correctement.

IRM avec biais

Nous avons pu voir dans la partie “Segmentation des coupes de l'IRM” que la segmentation sur des images nettes de coupe IRM nous donne un résultat cohérent. En réalité, une IRM donne un résultat pouvant être biaisé (Figure 9). Ce biais est dû à une homogénéité du champ magnétique \vec{B}_0 lors de l'acquisition.

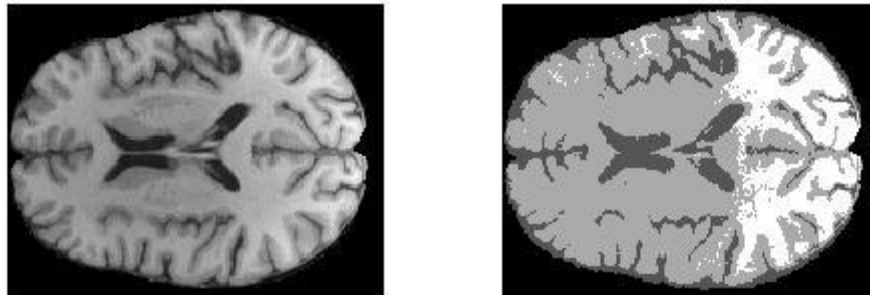


Figure 9 : Comparaison entre la coupe axiale $M0(:, :, 80)$ avec biais (à gauche) et sa segmentation en 4 classes (à droite)

On peut déjà remarquer sur cette Figure 9 qu'il y a un problème dans la segmentation de cette coupe du cerveau. On peut également le voir en fausse couleur:

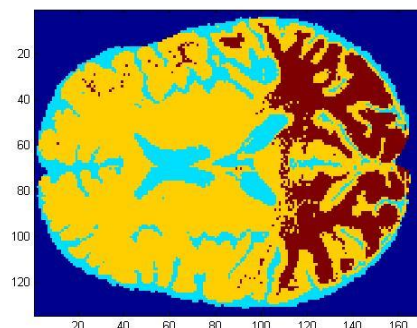


Figure 10 : Affichage en fausses couleurs de la segmentation de la coupe axiale avec biais $M0(:, :, 80)$

En comparant les segmentations d'une IRM sans biais (Figure 5) et une IRM avec biais (Figure 10), on remarque bien que les différentes composantes du cerveau (bleu ciel: liquide céphalo-rachidien, jaune : matière grise et marron: matière blanche) ne sont pas placées au bon endroit. On peut en déduire que c'est ce biais qui perturbe notre segmentation.

Pour mieux visualiser ce biais, nous avons utilisé la fonction *surf()* de Matlab. Cette fonction permet de représenter en 3D une image à partir de la 3e composante d'une matrice (ici le niveau de gris d'une image).

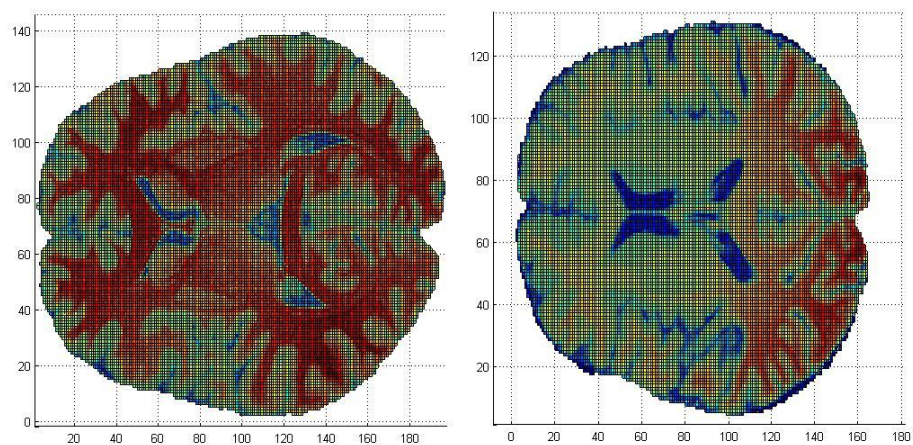


Figure 11 : Comparaison entre une coupe axiale $M0(:, :, 80)$ et une coupe axiale d'une IRM biaisé $M0(:, :, 80)$ avec la visualisation *surf()* de Matlab (vue du dessus)

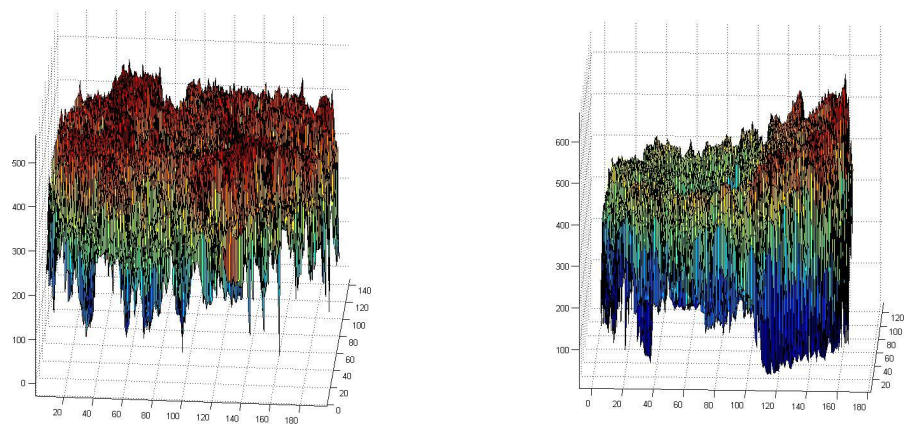


Figure 12 : Comparaison entre une coupe axiale $M0(:, :, 80)$ et une coupe axiale d'une IRM biaisé $M0(:, :, 80)$ avec la visualisation *surf()* de Matlab (vue de côté)

Grâce à ces représentations `surf()`, on peut mieux visualiser le biais de la seconde coupe axiale. Une partie des valeurs de cette coupe sont largement supérieures aux autres valeurs de cette même coupe.

Afin de corriger ce biais, nous allons chercher le plan qui approxime le mieux les valeurs des points de cette coupe axiale. Il suffira ensuite de soustraire à la valeur des points de la coupe la valeur correspondante sur le plan créé. Cela remettra l'ensemble des valeurs de notre coupe au même "niveau", et permettra de corriger le biais.

Régression par moindre carré

Pour corriger le biais, nous allons utiliser la méthode de régression par moindre carré. Pour ce faire nous avons premièrement calculé un plan qui approxime le mieux les valeurs des points de la coupe axiale de la Figure 12 (à droite). On a donc cherché le plan d'équation $z = ax + by + c$ qui approxime le nuage de points (fonction `surf()` de notre coupe axiale), c'est-à-dire qui minimise la somme des distances au carré entre chaque point et le plan.

En s'appuyant sur le fichier `plan_least_square.m`, nous avons redéfini les axes x, y et z de celui-ci comme suit :

- x : les numéros des lignes des points non nuls de notre nuage de points obtenus par la méthode `surf()` de Matlab.
- y : les numéros des colonnes des points non nuls de notre nuage de points
- z : la valeur en niveau de gris associée à la combinaison des deux premiers vecteurs de notre coupe axiale.

On obtient le résultat suivant :

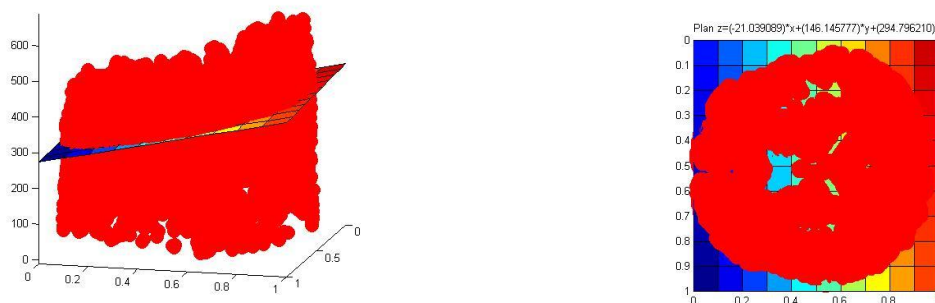


Figure 13 : Plan d'approximation créé à partir du nuage de points d'une coupe axiale M0(:,80) vue de côté (à gauche) et du dessus (à droite)

Afin de corriger le biais de notre coupe axiale, il nous suffit de soustraire à chacun des niveaux de gris de notre image, sa valeur correspondante sur le plan créé précédemment figure 14 (gauche).

Après la correction du biais on peut refaire notre segmentation, figure 14 (droite).

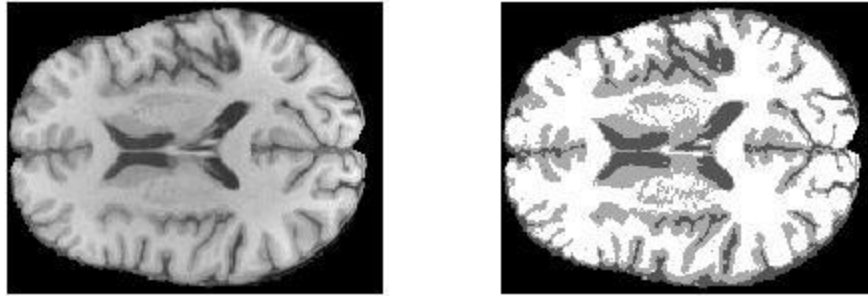


Figure 14 : Comparaison entre la coupe axiale $M0(:, :, 80)$ après correction du biais (à gauche) et sa segmentation en 4 classes (à droite)

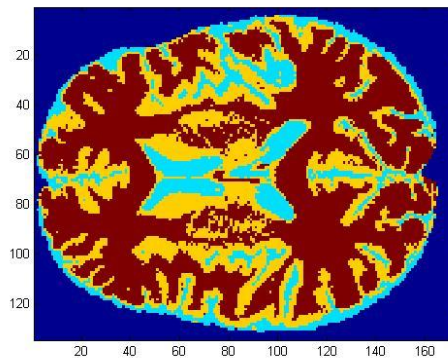


Figure 15 : Affichage en fausses couleurs de la segmentation de la coupe axiale $M0(:, :, 80)$ après correction du biais

La segmentation (figure 14 et 15) de la coupe axiale après correction du biais nous donne des résultats similaires à ceux obtenus sur la coupe sans biais (Figure 4 et 5). On peut en déduire que la correction du biais a été effective.

De plus, pour évaluer notre régression, nous pouvons de nouveau générer la surface corticale située entre le liquide céphalo-rachidien et la matière grise et la comparer à celle générée figure 8.

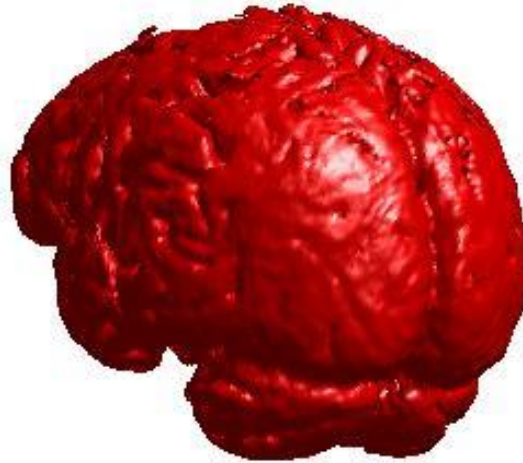


Figure 16 : Surface corticale du cerveau débiaisé à la coupe M0(:, :, 80)

Comme nous pouvons le constater, la figure 3D obtenue est légèrement plus lisse que celle générée figure 8 mais s'en rapproche suffisamment pour considérer que la régression appliquée est réussie.

Enfin, nous pouvons appliquer la correction du biais sur d'autres images que des images médicales comme le montre les figure 17 et 18.

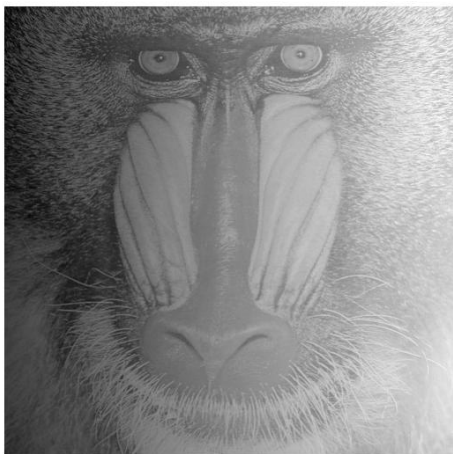


Figure 17 : Image biaisée

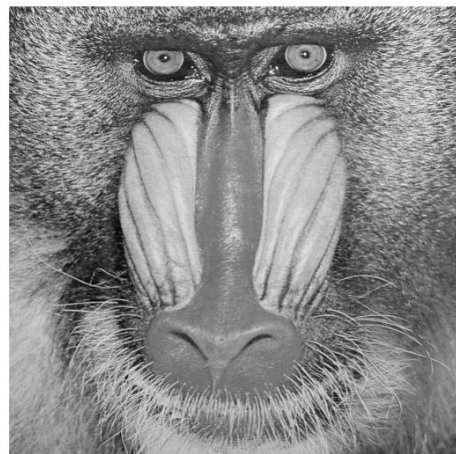


Figure 18 : Image débiaisée

Conclusion

L'intérêt de ce TP fut de pouvoir appliquer les algorithmes de visualisation et de segmentation vus en cours sur un volume 3D. Par ces biais, nous avons pu identifier les zones clés de certaines coupes du cerveau et en donner la représentation 3 dimensions. Au-delà de la compréhension du fonctionnement de ces outils, cela fut enrichissant car ce genre de dispositifs font partie des méthodes utilisées en recherches pour l'imagerie médicale. Il nous sera donc peut-être donné de travailler sur ces problématiques par la suite.