

TP2 - IMN530

FOUQUET, Jérémie et MÉTHOT, Vincent

17 mars 2014

Préface

Le code `python` utilisé dans ce TP est contenu dans le module `tp2` du dossier `Python`. Pour obtenir plus de renseignements sur `tp2`, l'utilisateur peut entrer la commande `tp2.fonction?` dans un terminal `ipython` et pour voir son code ; `tp2.fonction??`. Le code de la question 5 est contenu dans le script `ants.sh` du dossier `ANTS`. Le dossier `ANTS` contient aussi les deux images IRM du TP1 (`flair.nii` et `t1.nii`) ainsi que leurs images recalées. Le dossier `Data` contient les images 2D utilisées dans les questions 1-4.

Question 1 Histogramme conjoint

a Fonction `JointHist(I,J,nbin)`

La fonction `tp2.JointHist(I,J,nbin)` calcule l'histogramme conjoint de `I` et `J`.

b Somme des valeurs de l'histogramme conjoint

La fonction `tp2.verifSommeHisto(I,J,nbin)` retourne `True` si la condition

$$\sum_{k,l} H_{I,J}(k,l) = n \times p \quad (1)$$

est respectée, avec $n \times p$ les dimensions des images `I` et `J`. Peu importe les images `I` et `J` utilisées, si elles sont de même taille, la fonction `tp2.verifSommeHisto(I,J,nbin)` retourne `True`. Par exemple, `verifSommeHisto('Data/I1.png','Data/J1.png')` et `verifSommeHisto('Data/I2.png','Data/J2.png')` retournent `True`. Ceci vérifie bien l'éq. 1. Conceptuellement, cette condition s'explique par la définition de l'histogramme conjoint, qui recense le nombre de pixels d'une image à chaque intensité de l'image cible et source. La somme d'un histogramme conjoint donne donc le nombre de pixel d'une image.

c Analyse des histogrammes conjoints

Les histogrammes conjoints pour tous les couples d'images (I_k, J_k) de la fig. 1 sont présentés sur la fig. 2. Voici les principales observations faites pour chacun des couples :

1. La forte intensité de l'histogramme le long de la droite de pente unitaire et d'ordonnée à l'origine nulle indique que les valeurs des images sont très similaires et que celles-ci sont déjà pratiquement recalées. L'intensité décroissante et diffuse autour de la droite indique qu'au moins une des deux

images est entachée d'un bruit indépendant des intensités de l'autre image. L'écart-type du bruit semble constant en fonction de l'intensité de l'image.

2. La présence d'une droite de forte intensité indique qu'il existe une dépendance linéaire entre les valeurs de deux images et que celles-ci sont déjà pratiquement recalées. La pente différente de un indique qu'il existe un facteur d'échelle entre les valeurs des deux images (la gamme dynamique de l'image J est plus faible). L'ordonnée à l'origine non nulle montre que les valeurs de l'image J sont toutes majorées d'une constante par rapport à l'image I. Encore une fois, l'intensité décroissante et diffuse autour de la droite indique la présence de bruit indépendant dans au moins une des deux images.
3. L'absence de patron clair dans l'histogramme conjoint indique que les intensités des images ne sont en général pas reliées par une transformation évidente. La région agrandie montre que l'histogramme présente une forte intensité au point (255,255). Ceci signifie qu'une grande proportion des pixels des deux images situés aux mêmes endroits sont blancs (ont une intensité de 255).
4. L'intensité de l'histogramme conjoint est davantage concentrée autour d'une droite approximative, mais une région élargie autour de cette droite de même que d'autres régions ont aussi une intensité significative. Cela signifie que les deux images sont relativement semblables et recalées, mais que des diffaltes.
5. Les lignes droites observées montrent que des régions d'intensité constante dans l'image J chevauchent des régions d'intensités variables de l'image I. Il est difficile de dire à partir d'une inspection visuelle de l'histogramme si les images sont recalées.
6. Visuellement, l'histogramme a des caractéristiques très similaires à l'histogramme du couple 5. Cependant, les images sont davantage recalées (d'après une inspection visuelle des images). Cela illustre donc qu'il peut parfois être difficile de dire avec un seul coup d'oeil à l'histogramme conjoint si deux images sont recalées.

Question 2 Critère de similarité

a Somme des différences au carré

La fonction `tp2.SSD(I,J,nbin)` calcule la somme des différences au carré entre I et J.

b Coefficient de corrélation

La fonction `tp2.CR(I,J,nbin)` calcule le coefficient de corrélation entre I et J.

c Information mutuelle

La fonction `tp2.IM(I,J,nbin)` calcule l'information mutuelle entre I et J.

d Comparaison des trois critères

Les valeurs des trois critères de similarité présentés précédemment pour les 6 couples d'images fournis sont présentés au tab. 1.

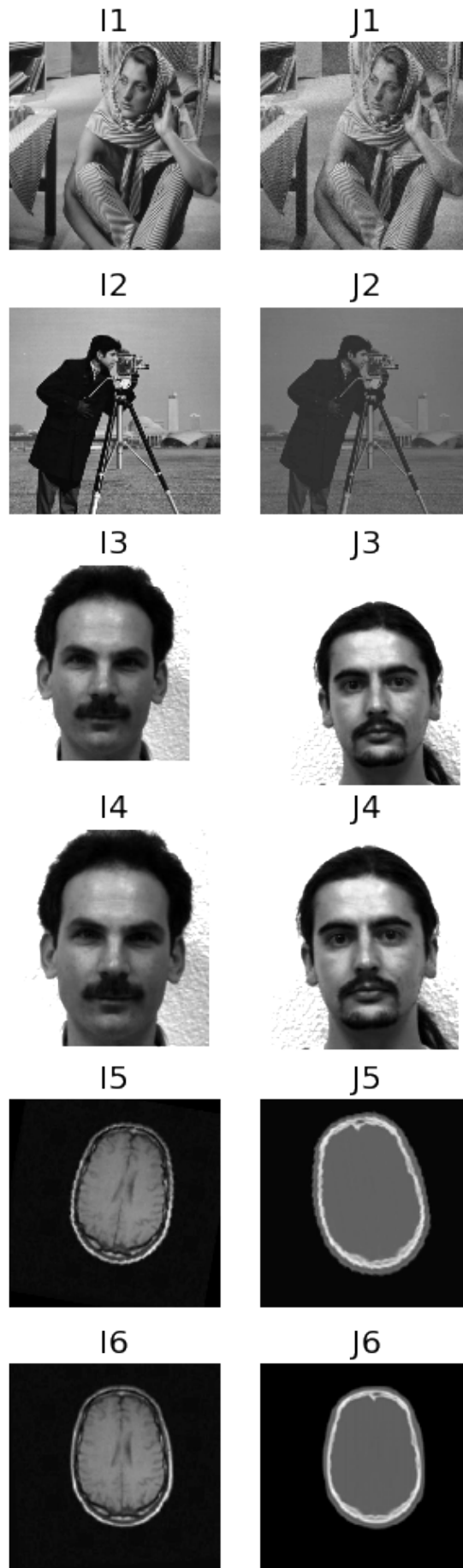


FIGURE 1 – Couples d'image (I_k, J_k) avec $k \in 1, 2, \dots, 6$.

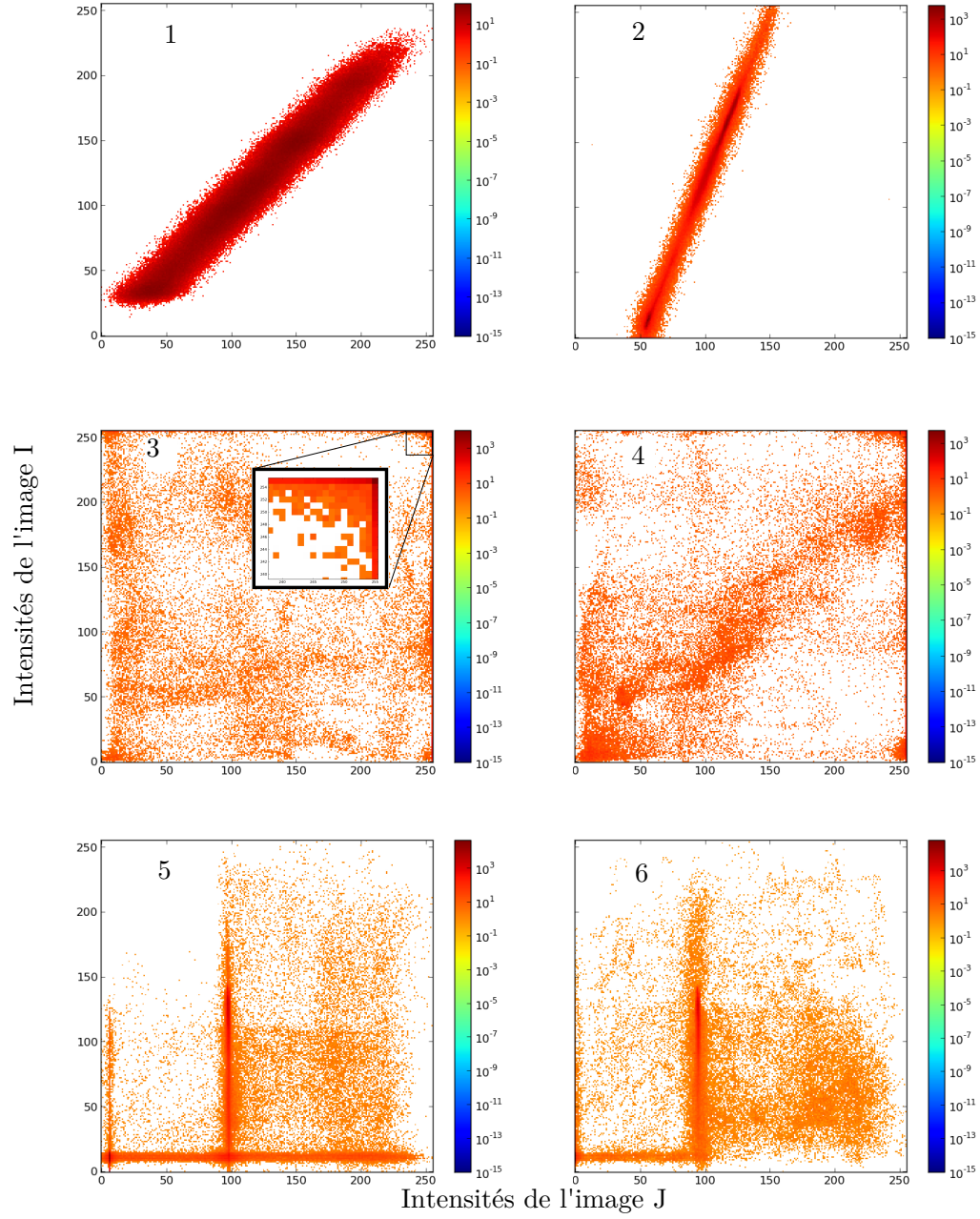


FIGURE 2 – Histogrammes conjoints pour les couples (I_k, J_k) avec $k \in 1, 2, \dots, 6$. Une échelle log a été utilisée pour tous les histogrammes.

k	SSD	CR	IM
1	16084808	0.978	1.554
2	30364219	0.996	2.228
3	4401503	0.143	0.668
4	3848301	0.564	1.023
5	20998301	0.656	0.555
6	42510000	0.780	0.675

TAB. 1 – Somme des différences au carré (SSD), Coefficient de corrélation (CR) et Information mutuelle (IM) pour les 6 couples d’images (I_k , J_k) avec 256 bin

La comparaison des SSD entre les couples d’image ne signifie pas grand chose, puisque ce critère n’est pas normalisé – il dépend des distributions d’intensité des deux images. Deux images très semblables, comme les couples $k = 1$ et $k = 2$, affichent donc des SSD très différentes.

L’inspection de la fig. 2 permet de prédire certains des critères de similarité. Les histogrammes 1 et 2 sont très linéaires; on s’attend donc à des coefficients de corrélation très près de 1, ce qui est le cas. Les histogrammes 4-6 semblent comporter des formes de lignes droites, ce qui est accentué par des coefficients de corrélation assez élevé. Le coefficient de corrélation pour $k = 3$, est très bas, ce qui s’explique par la translation entre les deux images, empêchant toute comparaison. De plus, l’information mutuelle est élevée même pour deux images différentes ($k = 4$) du moment qu’elles présentent des similarité structurelles et soient bien alignées. Ceci pourra s’avérer utile lors de recalage d’images de modalités différentes. Dernièrement, on voit que des translations et rotations réduisent fortement l’information mutuelle ($k = 3, 5, 6$) ce qui en fera un bon critère de recalage.

Question 3 Transformations spatiales

a Grille régulière de points 3D

La fonction `tp2.grille_test` génère une grille régulière de points 3d, lui fait subir une transformation affine et affiche les points initiaux en bleu ainsi que les points transformés en rouge.

b Transformation rigide

La fonction `tp2.trans_rigide(theta, omega, p, q, r)` retourne la matrice d’une transformation rigide en coordonnées homogènes. La figure 3 montre le résultat de cette fonction.

c Similitude

La fonction `tp2.similitude(s, theta, omega, p, q, r)` retourne la matrice d’une transformation de similarité en coordonnées homogènes. La figure 4 montre le résultat de cette fonction.

d Analyse des matrices de transformation

L’analyse visuelle des trois matrices ($M1$, $M2$ et $M3$) permet de déceler de quel genre de transformations il s’agit. Les matrices 3×3 en haut à gauche comporte les informations de rotation et dilatation. Ainsi, on pourrait trouver les valeurs propres ainsi que les vecteurs propres de cette sous-matrice pour

Transformation sur une grille de points

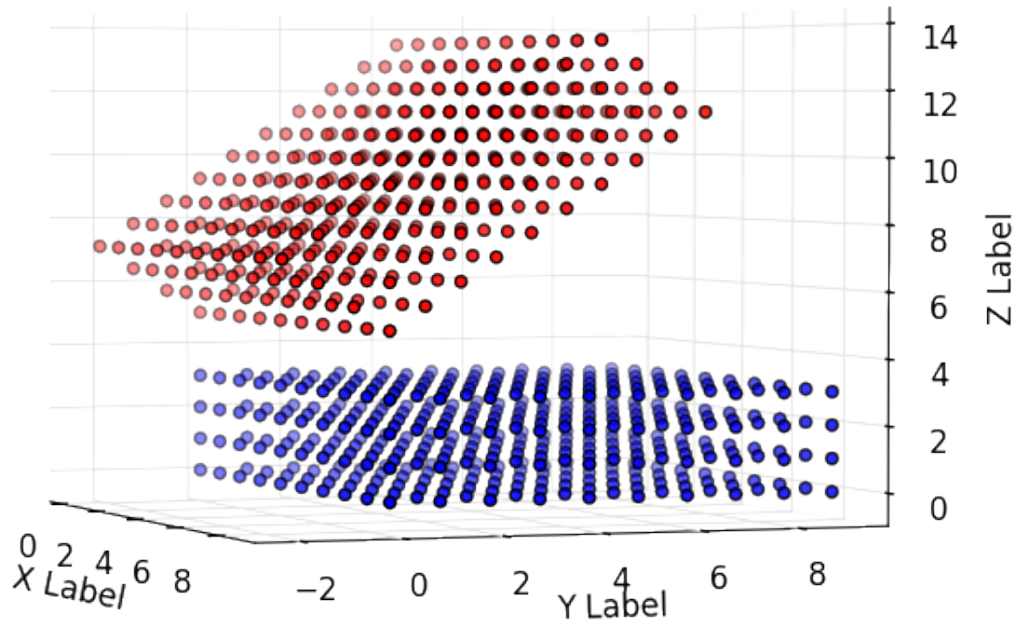


FIGURE 3 – Résultat de la commande
`tp2.grille_test(tp2.trans_rigide(pi/4, 0, 0, 0, 0, 5), 10, 10, 4).`

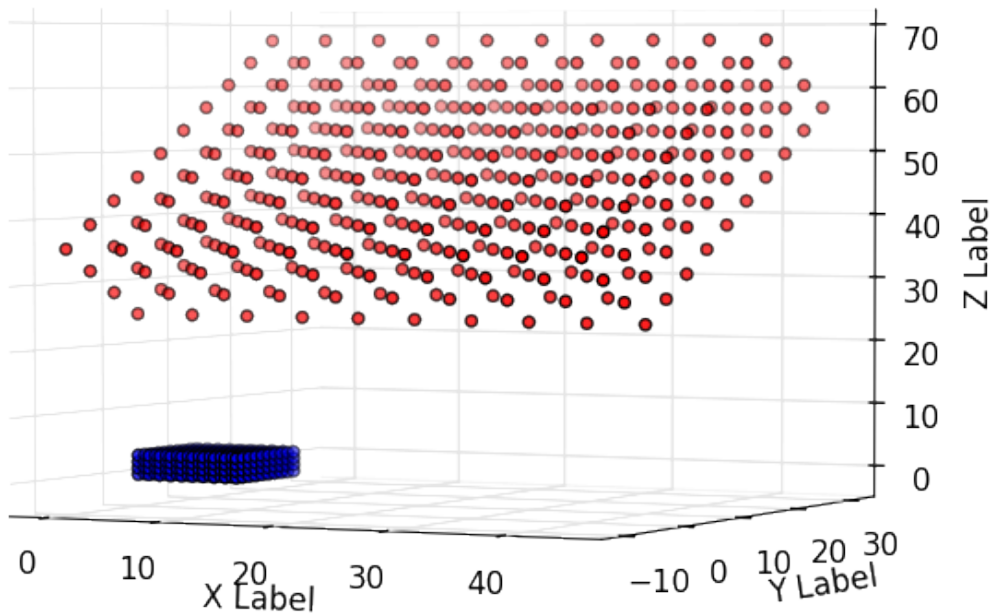


FIGURE 4 – Résultat de la commande
`tp2.grille_test(tp2.similitude(5, pi/4, 0, 0, 0, 0, 5), 10, 10, 4).`

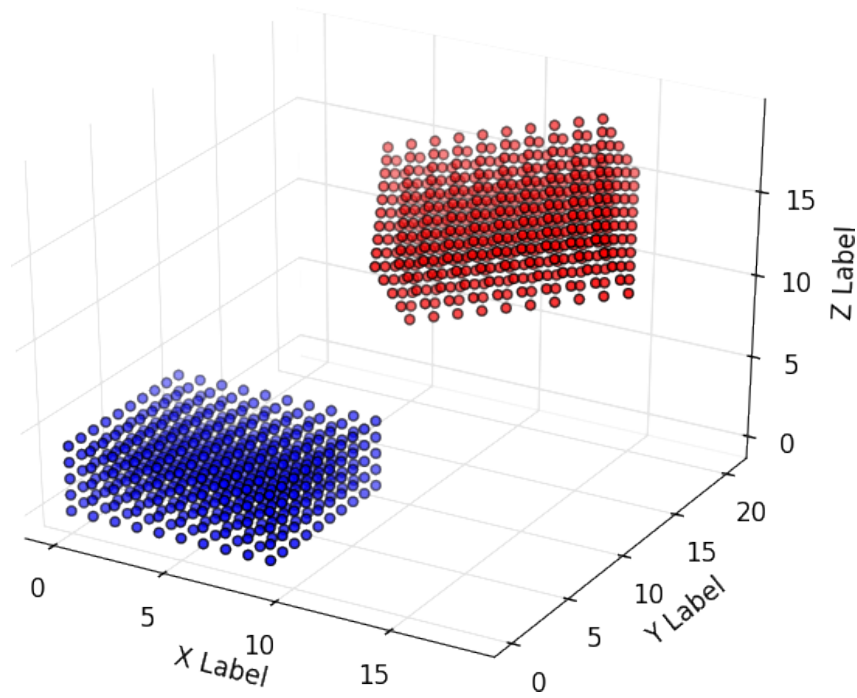


FIGURE 5 – Transformation M1 sur une grille de points

identifier les directions principales et les longueurs de la grille de points transformés. Nous contenterons de tracer les trois transformation à l'aide d'une grille de points régulière.

De plus, le vecteur 3 par 1 en haut à droite de la matrice donne la translation à appliquer après la rotation et la dilatation.

Ceci dit, la matrice M1 (fig. 5), comporte une rotation sans dilatation, suivit d'une translation de (10, 10, 10). La matrice M2 (fig. 6), rapetisse la grille, lui impose une rotation, puis une translation de (-3, 1.5, 0). Dernièrement, la matrice M3, (fig. 7) agrandit l'espace entre les points tout en donnant une rotation et une translation en (1.8115, 0.2873, 0.7404).

Question 4 Recalage iconique 2D simple

Pour les questions a à e, nous avons utilisé la descente de gradient à pas fixe pour minimiser la SSD. Nous mentionnons d'abord qu'il a été particulièrement difficile de trouver un critère d'arrêt efficaces pour des translations rigides quelconques. Nous avons essayé les critères suivants :

1. La norme du gradient est suffisamment petite pendant un certain nombre d'itérations
2. Les paramètres optimisés varient moins qu'un certain seuil pendant un nombre donné d'itérations
3. La SSD varie moins qu'un certain seuil pendant un nombre donné d'itération
4. La SSD ne fait qu'osciller autour d'une valeur moyenne
5. Un nombre d'itérations maximum est atteint

Parmi ces critères, aucun ne s'est avéré être efficace dans tous les cas de transformation rigide. Nous avons empiriquement choisi d'utiliser le critère 5 pour la question b, 3 pour la question d puis 3 pour la question e.

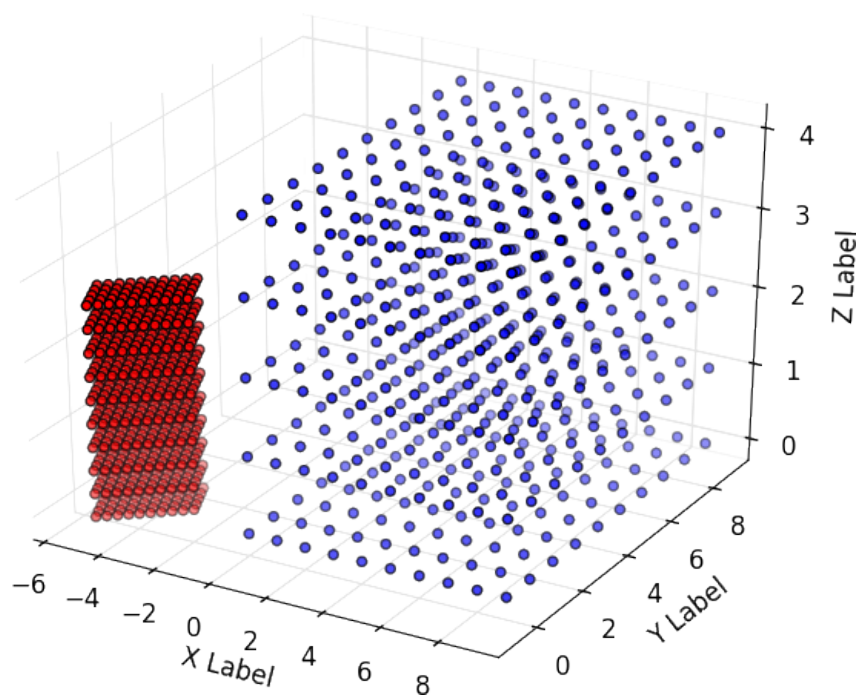


FIGURE 6 – Transformation M2 sur une grille de points

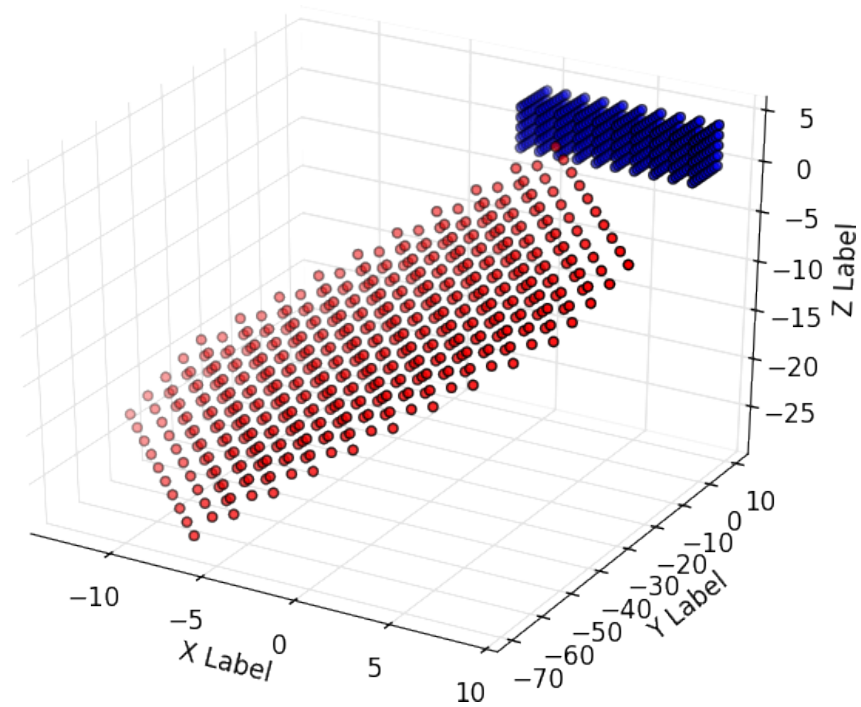


FIGURE 7 – Transformation M3 sur une grille de points

	Im. 1	Im. 2	Im. 3
$(p, q)_{ini}$ [pix]	(26.5644, 45.9758)	(-0.1177, 19.2243)	(9.2674, 5.7847)
$(p, q)_{rec}$ [pix]	(-26.5791, -45.9541)	(0.1024, -19.2381)	(-9.2521, -5.7807)

TAB. 2 – Pour 3 exemplaires de l’image Brain_MRI_1 initialement translatées par les vecteurs $(p, q)_{ini}$, les vecteurs de translation $(p, q)_{rec}$ ont été obtenus grâce au recalage par descente de gradient à pas fixe présenté à la [Question 4 b](#)

a Fonction translation(I,p,q)

Voir la fonction `translation(I,p,q)` du module `tp2`.

b Recalage d’images translatée

La fonction `rec2dtrans(I,J)` permet de recaler I sur J en ne faisant que translater I.

La fonction `num4b()` permet de générer 3 translations aléatoires de l’image Brain_MRI_1 puis de les recaler sur l’image de départ en utilisant la fonction `rec2dtrans(I,p,q)`.

Les résultats ainsi obtenus sont présentés dans le tab. 2 (vecteurs translations initiaux et de recalage), dans la fig. 8 (évolution de la SSD en fonction du nombre d’itérations) et dans la fig. 9 (évaluation visuelle du recalage). Les paramètres qui ont été utilisés pour les recalages sont les valeurs par défaut de la fonction `num4b()`.

Sur le graphique de la SSD (fig. 8), un premier élément à remarquer est le nombre élevé d’itérations effectuées vers la fin du recalage alors que la SSD était quasiment constante. Cela est dû au fait que nous avons eu de la difficulté à trouver un critère d’arrêt efficace et que nous avons donc choisi d’arrêter la descente après un nombre d’itérations fixe (500). On remarque aussi que la SSD a eu le temps d’atteindre un plateau constant avant la fin du recalage, ce qui indique que nous avons très probablement convergé vers un minimum. Enfin, on voit aussi clairement que la courbe de la SSD n’est pas strictement décroissante. Cela est principalement dû au pas de gradient qui pouvait parfois être trop grand et donc mener à une région où la SSD était plus élevée. Pour corriger cela, il faudrait utiliser un pas de gradient adaptatif plutôt que fixe. Cela peut se faire avec un *line search* ou avec la méthode de Newton entre autres.

Le tab. 2 de même que la fig. 9 montrent que le recalage a convergé vers des valeurs acceptables du vecteur de translation.

c Fonction rotation(I,theta)

Voir la fonction `rotation(I,theta)` du module `tp2`.

d Recalage d’images rotatées

La fonction `rec2drot(I,J)` permet de recaler I sur J en ne faisant que rotater I.

La fonction `num4d()` permet de générer 3 rotations aléatoires de l’image Brain_MRI_1 puis de les recaler sur l’image de départ en utilisant la fonction `rec2drot(I,p,q)`.

Les résultats obtenus sont présentés dans le tab. 3 (angles initiaux et de recalage), dans la fig. 10 (évolution de la SSD en fonction du nombre d’itérations) et dans la fig. 11 (évaluation visuelle du recalage). Les paramètres qui ont été utilisés pour les recalages sont les valeurs par défaut de la fonction `num4d()`.

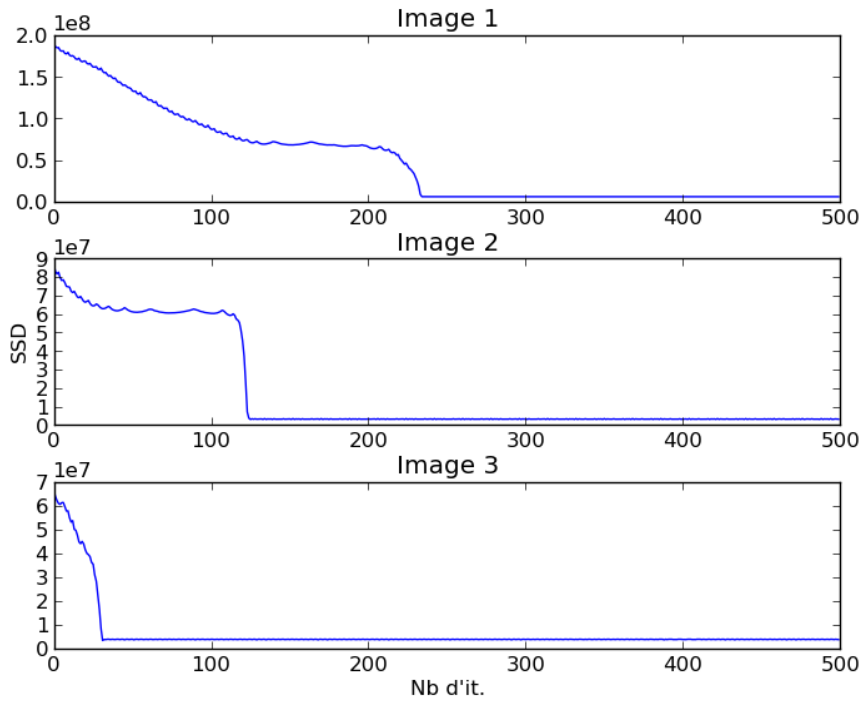


FIGURE 8 – Évolution de la SSD lors de trois recalages effectués avec **rec2dtrans**. Trois translations de l'image Brain_MRI_1, décrites par les vecteurs $(p, q)_{ini}$ du tab. 2, ont été recalées sur l'image initiale Brain_MRI_1.

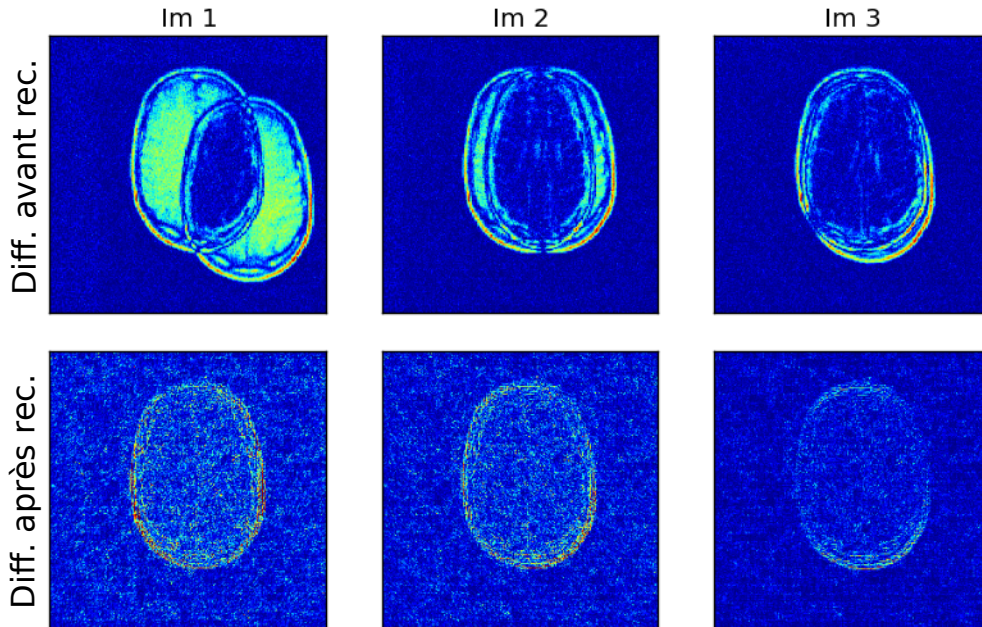


FIGURE 9 – Différence absolue entre l'image cible et l'image source avant et après recalage pour trois recalages effectués avec la fonction **rec2dtrans**. Trois translations de l'image Brain_MRI_1, décrites par les vecteurs $(p, q)_{ini}$ du tab. 2, ont été recalées sur l'image initiale Brain_MRI_1.

	Im. 1	Im. 2	Im. 3
θ_{ini} [rad]	-0.0652	0.1730	-0.3926
θ_{rec} [rad]	0.0652	-0.1730	0.3926

TAB. 3 – Pour 3 exemplaires de l’image Brain_MRI_1 initialement rotatés par les angles θ_{ini} , les angles θ_{rec} ont été obtenus grâce au recalage par descente de gradient à pas fixe présenté à la [Question 4 d](#).

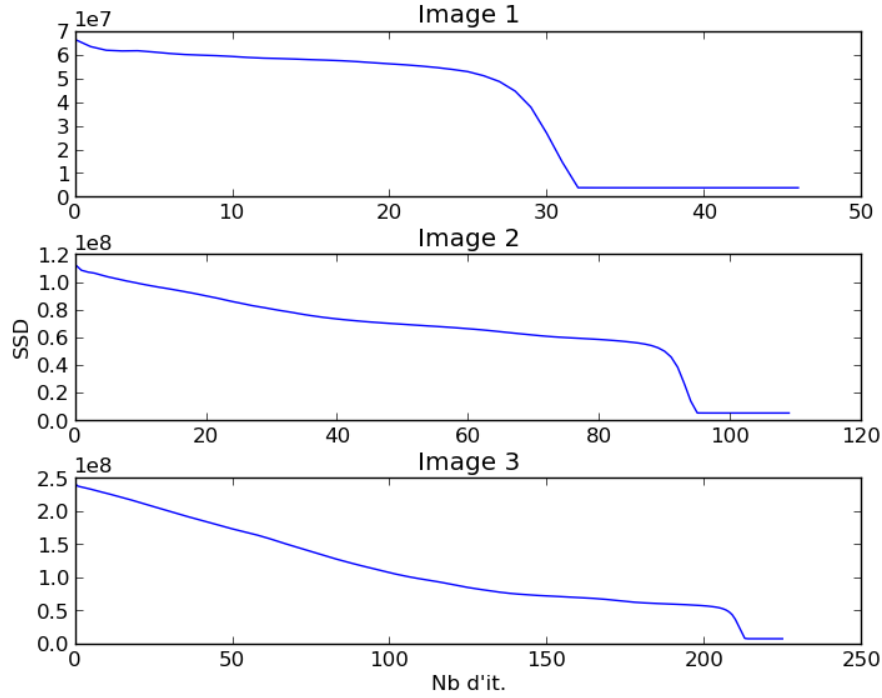


FIGURE 10 – Évolution de la SSD lors de trois recalages effectués avec `rec2drot`. Trois rotations de l’image Brain_MRI_1, décrites par les angles θ_{ini} du tab. 3, ont été recalées sur l’image initiale Brain_MRI_1.

Sur le graphique de la SSD (fig. 10), on remarque que la courbe est plus lisse que sur la fig. 8 et que les régions où la SSD ne décroît pas de façon monotone sont plus rares. Cela indique probablement que le pas de gradient utilisé était suffisamment petit. L’utilisation d’un pas adaptatif aurait pu accélérer la méthode.

Le tab. 3 de même que la fig. 11 montrent que le recalage a convergé vers des valeurs acceptables d’angles de rotation.

e Descente de gradient pour transformation rigide générale

La fonction `rec2dpasfixe(I, J)` permet de recaler I et J en considérant les transformations rigides générales. Le recalage des images Brain_MRI_2,3,4 sur Brain_MRI_1 a donné les valeurs de (p, q) et de θ présentées dans le tab. 4. Les recalages sont aussi montrés sur la fig. 12.

Il apparaît que les recalages de Brain_MRI_2,3 ont fonctionné, mais pas celui de Brain_MRI_4. Ceci est probablement dû à un minimum local, dans lequel nous sommes tombés lors de la descente de gradient.

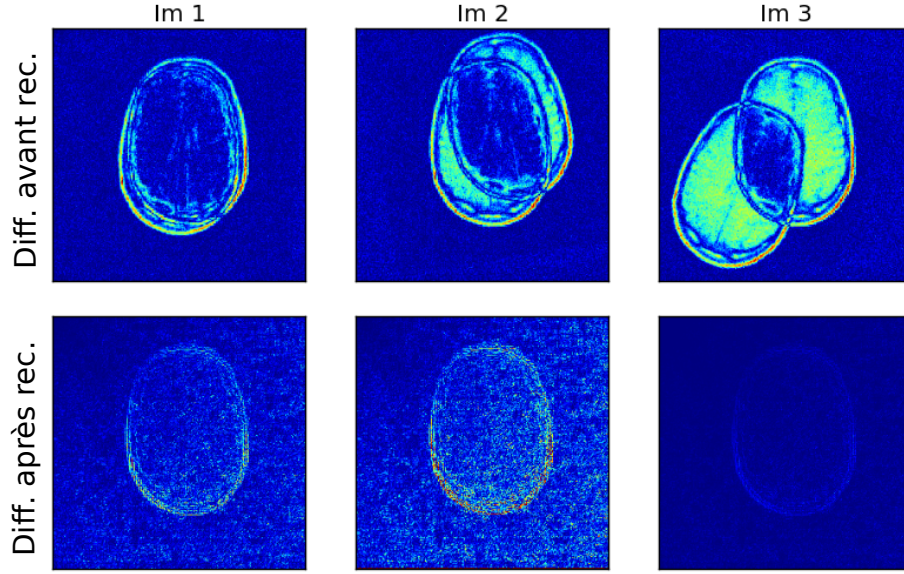


FIGURE 11 – Différence absolue entre l'image cible et l'image source avant et après recalage pour trois recalages effectués avec la fonction `rec2drot`. Trois rotations de l'image `Brain_MRI_1`, décrites par les angles θ_{ini} du tab. 3, ont été recalées sur l'image cible `Brain_MRI_1`.

	Brain_MRI_2	Brain_MRI_3	Brain_MRI_4
(p, q) [pix]	$(-23.08, -6.775)$	$(-4.28, 5.67)$	$(-8.18, -12.78)$
θ [rad]	0.014	0.17	-0.02

TAB. 4 – Paramètres des transformations rigides obtenues avec `rec2dpasfixe` lors du recalage des images `Brain_MRI_2,3,4` sur `Brain_MRI_1`

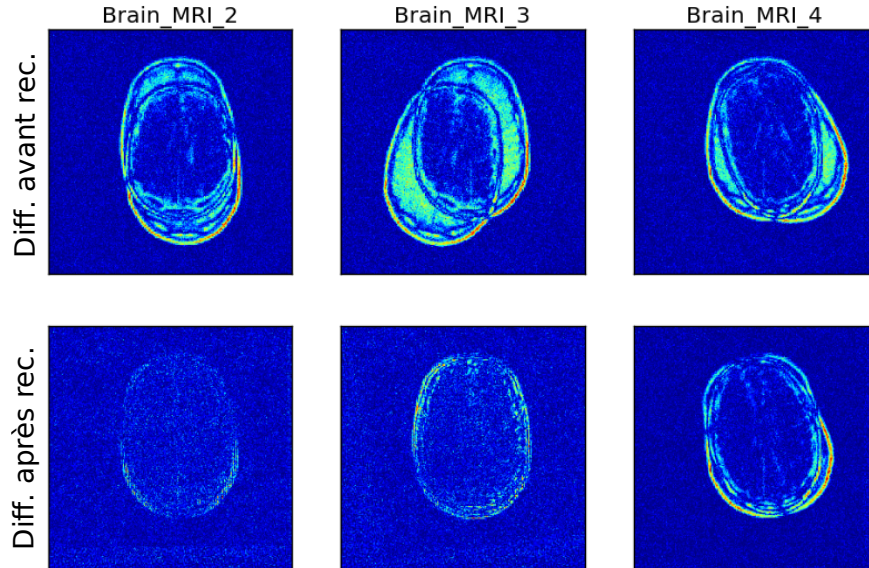


FIGURE 12 – Recalage des images `Brain_MRI_2,3,4` sur `Brain_MRI_1`. L'image 4 n'a pas été bien recalée.

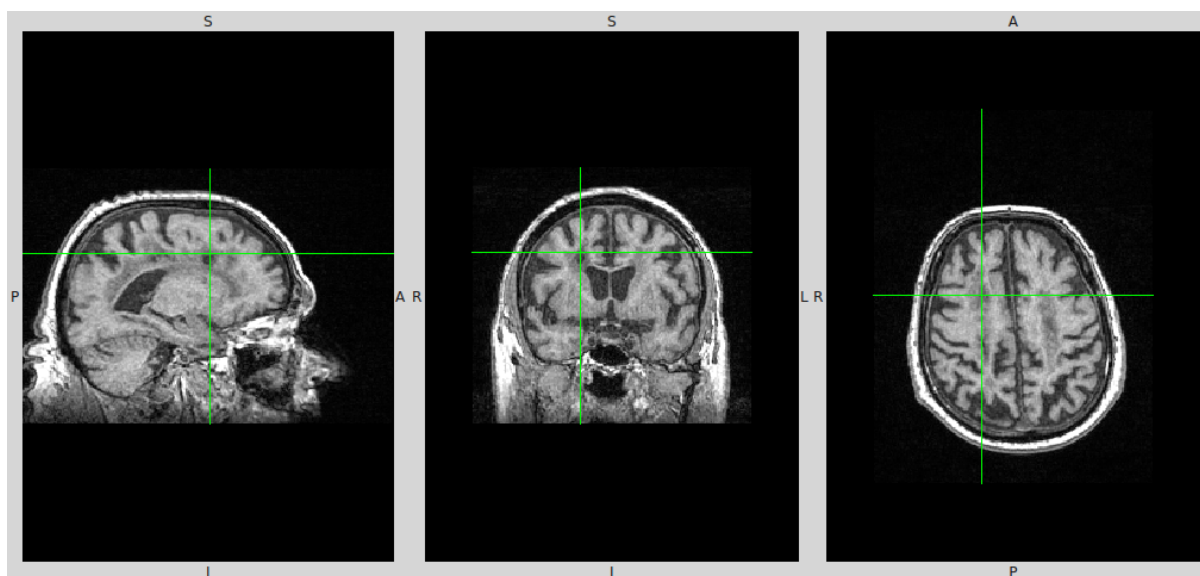


FIGURE 13 – Image t1.

e.1 Descente de gradient à pas adaptatif

Pour améliorer la descente de gradient à pas fixe, nous avons essayé d'implémenter un *line search*, ce qui apparaît dans la fonction `rec2doptimize`. Cependant, nous n'avons pas réussi à faire fonctionner la fonction `scipy.optimize.line_search` (voir ligne commentée contenant cette fonction). Nous avons donc essayé d'implémenter une méthode s'approchant vaguement du line search, soit : tant que le SSD après le pas de gradient $>$ SSD avant le pas de gradient, on diminue d'un certain facteur le pas pour l'itération en cours. Cette méthode était parfois intéressante (par exemple pour l'image `Brain_MRI_3`), mais il n'était pas évident de dire si elle était vraiment supérieure à la méthode du pas fixe.

Question 5 Recalage non-linéaire

Les images `t1.nii` (fig. 13) et `flair.nii` proviennent du même sujet, mais de deux modalités d'IRM différentes. De plus, la résolution des deux images est extrêmement différente, la première étant isotrope et la seconde étant fortement anisotrope. Un recalage linéaire utilisant une bonne méthode d'interpolation ainsi qu'un critère d'Information Mutuelle pourrait donc être une bonne solution (fig. 14), toutefois les différences de forme entre les deux modalités pourraient causer des différences qu'il serait impossible de prendre en compte sans un recalage non-linéaire (fig. 15).

On a utilisé `fsl flirt` pour le recalage linéaire (avec information mutuelle, et 6 degrés de liberté). On a comparé le recalage linéaire de `flirt` avec le recalage non-linéaire de ANTS (information mutuelle, multi-résolution et régularisation gaussienne). Les commandes utilisées et les images obtenues se retrouvent dans le dossier ANTS.

Les deux méthodes de recalage semblent, au premier abord, fournir un recalage des images permettant d'identifier et de comparer les structures cérébrales identiques. Toutefois, le recalage non-linéaire semble comporter moins d'artéfact dû à l'anisotropie et au changement de contraste, permettant par exemple d'identifier plus précisément un « trou » près du ventricule gauche.

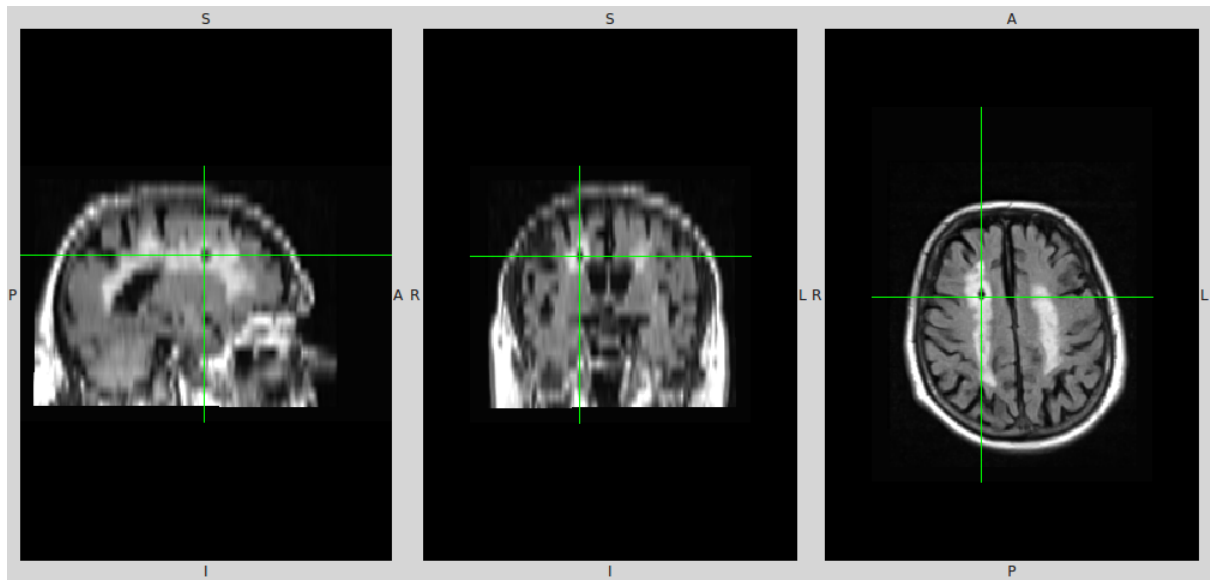


FIGURE 14 – Image flair recalée sur t1 par flirt.

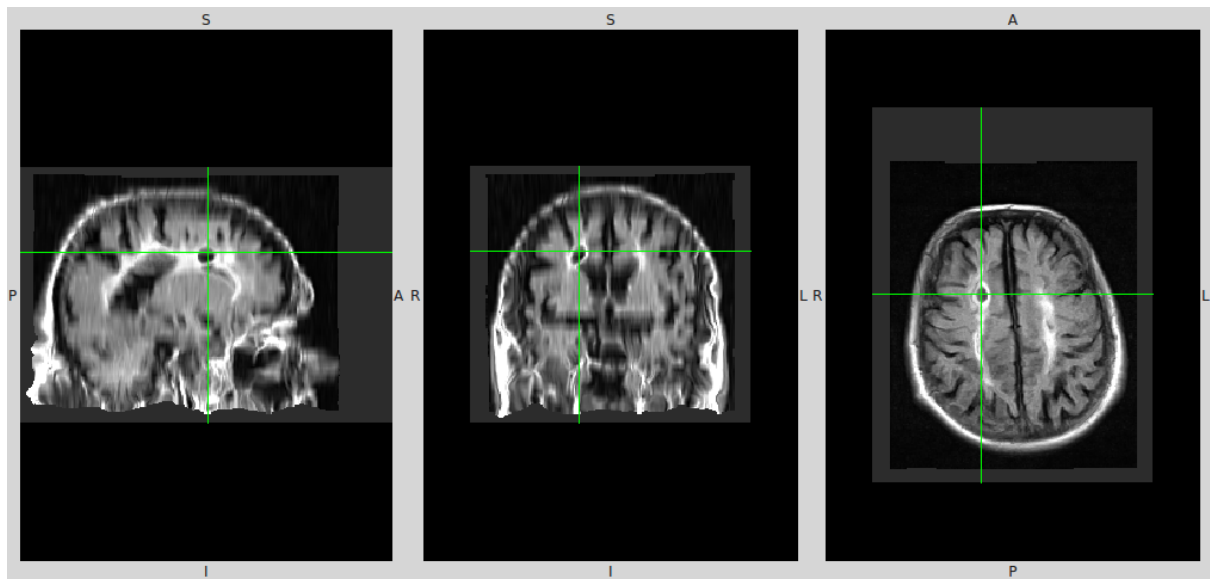


FIGURE 15 – Image flair recalée sur t1 par ANTS.