



Rapport VTK/IKT

Groupe:

W

Membres du groupe:

Hicham Selkti, Thomas Vasseur, Kenza Azennoud

Majeur IMAGE

Promo 2025

July 2024

Contents

1	Introduction	3
2	Données	3
3	Explication des choix techniques	3
3.1	Partie Recalage	3
3.2	Partie Visualisation	3
3.3	Partie Segmentation	4
4	La limite des techniques utilisées	4
4.1	Partie Recalage	4
4.2	Partie Visualisation	4
4.3	Partie Segmentation	4
5	Résultats et analyses	5
5.1	Step 1:	5
5.2	Step 2:	6
5.3	Step 3:	7
6	Conclusion	9

1 Introduction

Ce projet a pour objectif de réaliser le suivi des changements d’une tumeur à partir de deux scans effectués sur un même patient à des dates différentes, en utilisant les bibliothèques ITK (Insight Segmentation and Registration Toolkit) et VTK (Visualization Toolkit) pour effectuer le recalage des volumes, la segmentation des tumeurs et la visualisation des changements entre les deux tumeurs.

2 Données

- case6_gre1.nrrd
- case6_gre2.nrrd

3 Explication des choix techniques

3.1 Partie Recalage

- *Type de Transformation:* Nous avons utilisé Euler3DTransform, afin de pouvoir aligner des images volumétriques en 3D.
- *Optimization method:* Nous avons choisi GradientDescentOptimizer, pour converger rapidement vers le minimum local.
- *La Métrique:* Mean Squares, parce qu’elle permet de minimiser l’impact des erreurs expérimentales, en évaluant la similarité entre les images recalées et en maximisant la corrélation entre les intensités des voxels.
- *Resampling:* Nous avons utilisé ResampleImageFilter pour appliquer la transformation de recalage à l’image de case6_gre1.nrrd afin de produire une image recalée alignée avec l’image de case6_gre2.nrrd.

3.2 Partie Visualisation

- *Conversion de l’image ITK en image VTK:* Nous avons utilisé la fonction `array_from_image` d’ITK pour convertir l’image ITK en un tableau NumPy, permettant de manipuler facilement les données de l’image. Ensuite, nous avons utilisé la fonction `numpy_to_vtk` de `vtk.util.numpy_support` pour transformer le tableau NumPy en un tableau VTK. Puis, nous avons créé une image volumétrique en utilisant `vtkImageData` et avons défini ses paramètres en fonction de l’image ITK. Finalement, nous avons assigné les données de notre tableau VTK aux scalaires de l’image VTK, permettant de représenter les valeurs d’intensité des voxels.

- *Interaction:* Nous avons configuré des callbacks pour permettre l'interaction avec les tranches à l'aide de la souris. En positionnant le curseur sur l'image, puis en cliquant sur le bouton gauche et en effectuant un mouvement vertical (de haut en bas et vice versa), cela permet d'explorer différentes tranches des volumes.

3.3 Partie Segmentation

- Nous avons choisi d'utiliser la segmentation basée sur la croissance de région. Nous avons choisi cette méthode car nous travaillons sur des tumeurs, et cela nous semblait plus facile comme premier essai de segmentation d'utiliser une méthode semi-automatique où nous pouvons choisir les éléments visuellement en indiquant nous-mêmes un point sur les images.

4 La limite des techniques utilisées

4.1 Partie Recalage

- *Le choix des paramètres:* Il est important de choisir correctement les paramètres de l'optimiseur (LearningRate, MinimumStepLength, NumberOfIterations). Une mauvaise configuration peut conduire à une convergence lente ou à un recalage incorrect.
- *Le choix de la métrique:* La métrique Mean Squares peut être sensible aux variations d'intensité entre les images. Cela peut poser problème si les images à recaler ont des niveaux de contraste très différents.

4.2 Partie Visualisation

- *L'utilisation des tableaux NumPy pour la conversion ITK vers VTK:* Lors de la conversion de NumPy à VTK, il peut y avoir une perte de précision des données, ce qui peut affecter les résultats de l'image par la suite.
- *L'utilisation de l'interaction:* le déplacement vertical des tranches avec la souris n'est pas intuitif pour tout le monde, il est donc nécessaire de le préciser à chaque fois.

4.3 Partie Segmentation

- *Limitations de la croissance de région :* Cette technique fonctionne bien sur une image 2D où le point d'intérêt est fixe, mais sur les images en 3D, il peut y avoir des variations, notamment lorsqu'on modifie la profondeur. Cela peut entraîner une perte de la segmentation de la tumeur ou la segmentation d'un mauvais objet (comme un os).

5 Résultats et analyses

5.1 Step 1:

Avant de procéder au recalage, nous avons essayé de visualiser nos données sans modification sous différentes vues en utilisant `vtkNrrdReader`. On peut voir les tranches en déplaçant la souris de gauche à droite. Pour voir cette étape, vous pouvez exécuter le fichier `visu.py`.

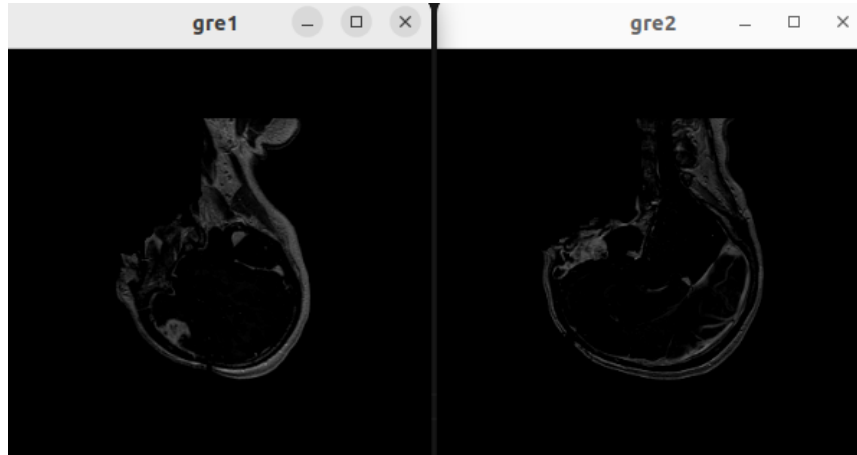


Figure 1: Vue axial

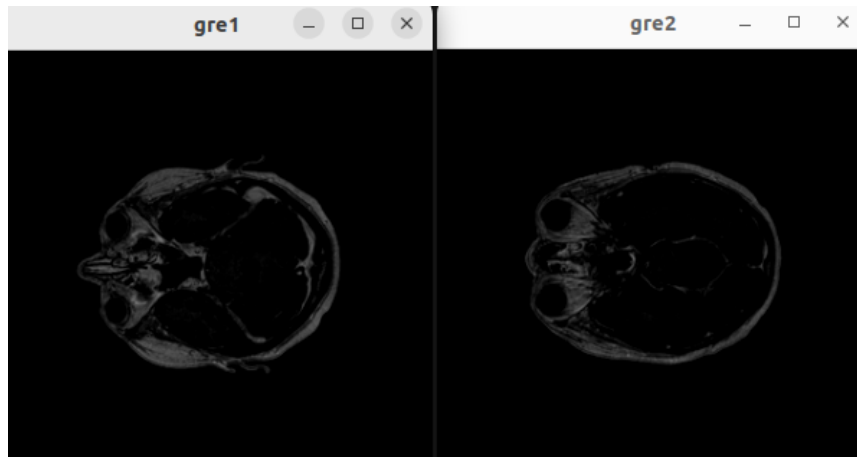


Figure 2: Vue coronal

Comme on peut le voir, il y a une grande différence entre `case6_gre1.nrrd` et `case6_gre2.nrrd`.

Ces résultats nous ont permis de constater à l'œil nu qu'il y a effectivement une différence entre les deux ensembles de données.

5.2 Step 2:

Nous avons ensuite ajouté la méthode de recalage pour aligner les deux images et comparer les changements de manière plus précise. Pour aligner les images, nous avons utilisé (comme mentionné dans la partie Explication des choix techniques) les paramètres suivants :

- Transformation : Euler3DTransform
- Optimiseur : GradientDescentOptimizer, avec les paramètres suivants:

```
optimizer.SetLearningRate(4.0)
optimizer.SetMinimumStepLength(0.01)
optimizer.SetNumberOfIterations(200)
```
- Métrique : MeanSquares
- Filtre de rééchantillonnage : ResampleImageFilter

Voici le resultat qu'on obtient:

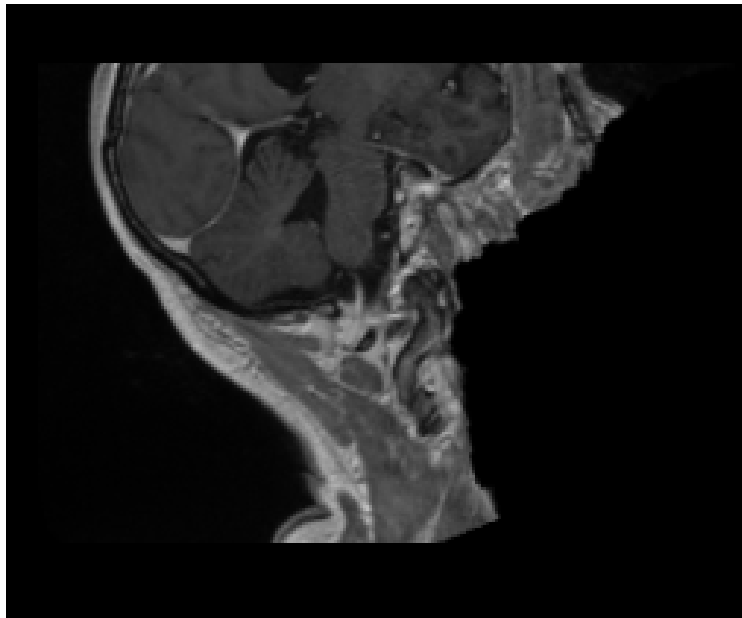


Figure 3: Image après recalage

Ensuite nous avons ajouté des interactions permettant de bouger la souris de manière verticale pour voir les différentes tranches.

Le recalage a permis d'aligner correctement les deux ensembles de données, on peut visualisé cela grace au resultat qu'on a obtenu, ce qui nous permet de faire une comparaison plus précise des changements dans la tumeur.

5.3 Step 3:

Nous avons ensuite effectué la segmentation des tumeurs à l'aide de la méthode basée sur la croissance de région. Pour cela, nous avons réalisé dans un premier temps, une normalisation de l'image afin que les valeurs soient comprises entre 0 et 255, puis nous avons appliqué une méthode de lissage pour réduire le bruit.

Nous n'étions cependant pas entièrement sûrs de nos résultats de recalage faisant que nous n'étions pas sûrs des points que l'on devait choisir pour réaliser la croissance de région. Nous avons donc décidé de réaliser la segmentation sur les images d'origine non recalées pour s'assurer que les coordonnées des points d'intérêt soient correctes.

Pour la partie segmentation au lieu de travailler directement en 3D, nous avons segmenté chaque tranche individuellement en 2D puis reconstruit l'image 3D. Bien que cette approche ne soit pas idéale, elle permet de contourner certaines difficultés associées à la segmentation 3D.

Paramètres utilisés pour la segmentation :

- Filtre de diffusion anisotrope:

```
GradientAnisotropicDiffusionImageFilter.New(Input=img,  
NumberOfIterations=20, TimeStep=0.04, ConductanceParameter=3)
```

- pour case6_gre1.nrrd:

```
seedX = 125,  
seedY = 65,  
low_threshold= 110.,  
high_threshold=150.
```

- pour case6_gre2.nrrd:

```
seedX = 125,  
seedY = 80,  
low_threshold= 110.,  
high_threshold=170.
```

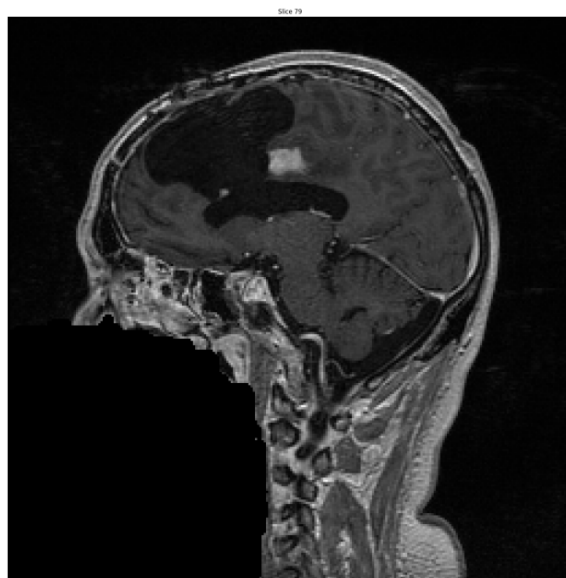


Figure 4: Slice du set1

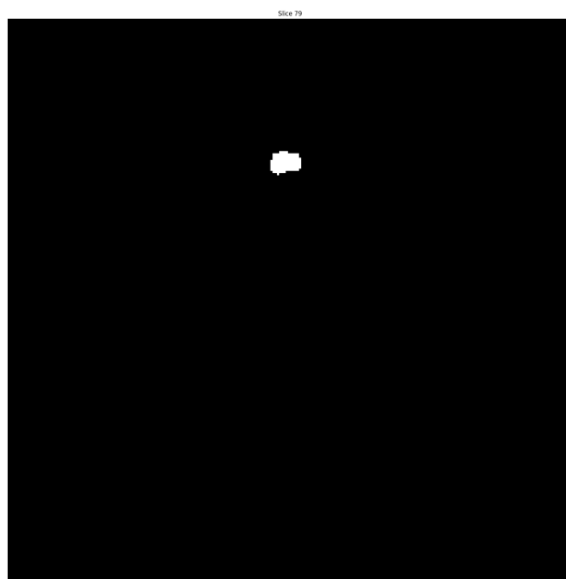


Figure 5: La tumeur

Les résultats de la segmentation sont assez satisfaisants. Cependant, nous n'avons pas réussi à ajouter des interactions pour la partie segmentation.

Vous pouvez visualiser nos résultats de la segmentation en exécutant le fichier `seg.py`.

6 Conclusion

Ce projet a permis de mettre en pratique les bibliothèques ITK et VTK pour réaliser une étude de l'évolution d'une tumeur à partir de deux scans effectués sur un même patient à des dates différentes. Nous avons d'abord visualisé les images brutes grâce à VTK, puis nous avons utilisé ITK pour aligner les deux images et effectuer la segmentation. Nous avons également ajouté des interactions pour permettre une exploration interactive des tranches, et finalement visualisé nos images grâce à VTK.

En conclusion, ce projet a démontré l'efficacité des outils ITK et VTK pour le recalage, la segmentation et la visualisation d'images volumétriques.

Voici le lien vers notre repo github: [Projet-vtk-github](#)