11763 - Medical Image Processing

Final Project:

DICOM Loading, Visualization and 3D Coregistration

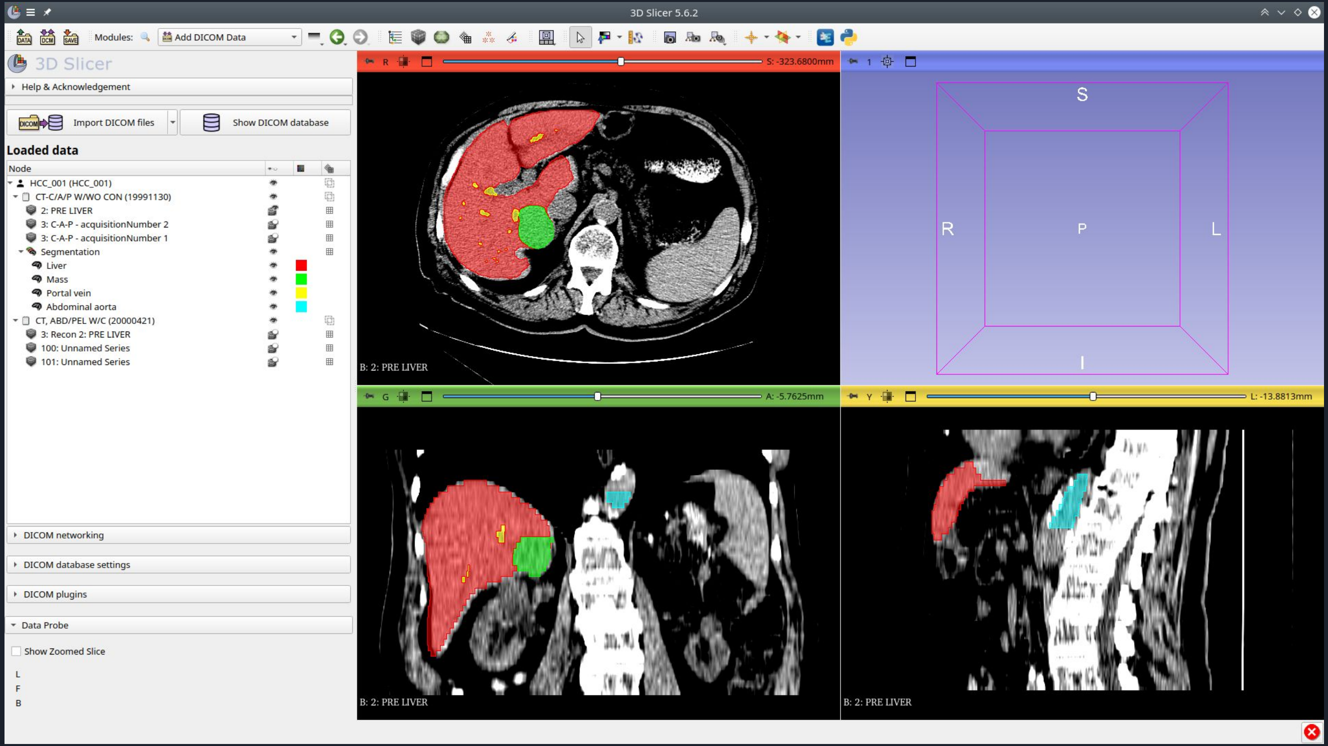
Guillem Bibiloni Femenias

Contents

1.

DICOM Loading and Visualization

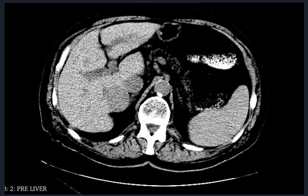
1.1 3D Slicer



Headers: Slice Location, Slice Thickness, Acquisition Number

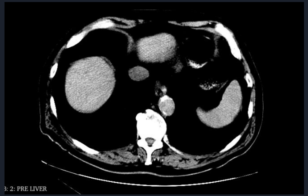
1.2 Image Rearrangement

Pixel Array Slice Location

1-03.dcm ( , 21)

1-05.dcm ( , 5)

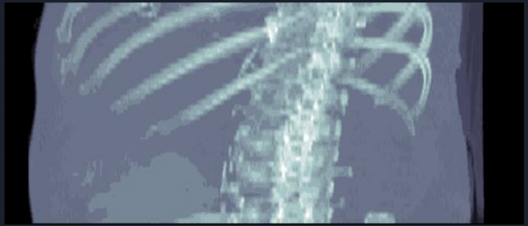
• • •

1-31.dcm ( , 37)

Sort By
Slice Location

And stack
them all!

3D CT Image

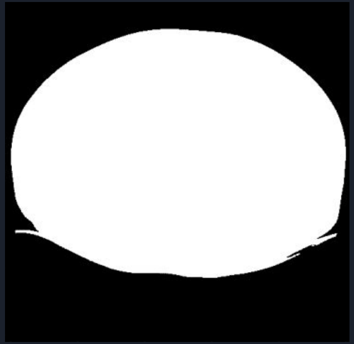


1.3 Artifact Removal

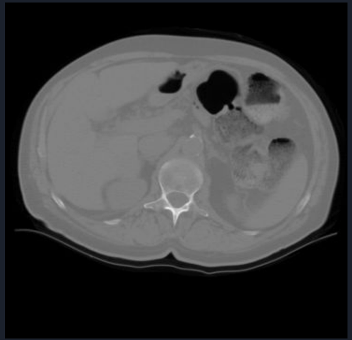
Mean through axis = 0

Binarize

Biggest Contour



Use it as a mask

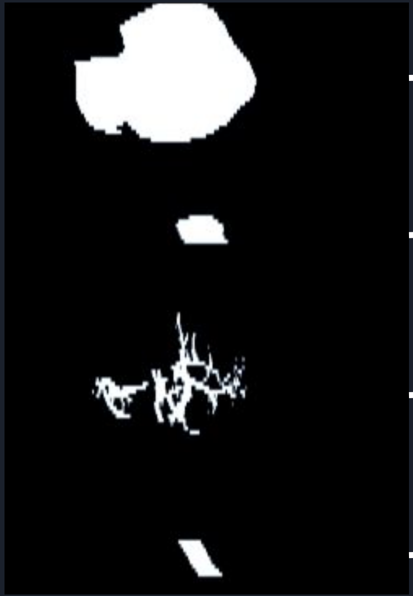


if $p < 0$ → $p := 0$

1.4 Segmentation

Arrange slices according to ImagePositionPatient and SliceLocation

ImagePositionPatient



1-1.dcm

[..., ..., -248]

...

[..., ..., -323]

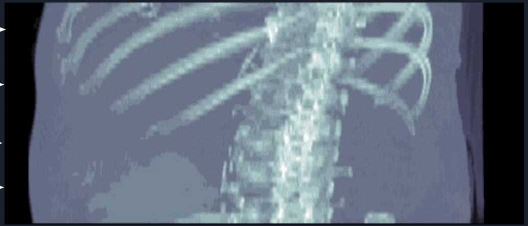
...

[..., ..., -392]

...

[..., ..., -402]

3D CT Image

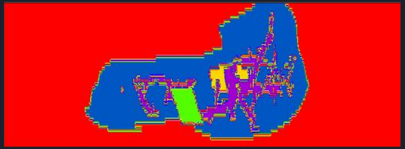


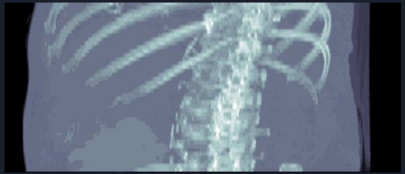
1.5 GIF animation

Rotate on axial plane
both the segmentation and CT image.

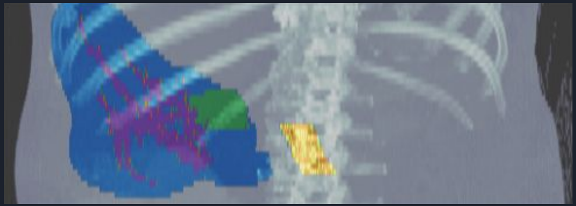
Adjust CT image
and segmentation

Maximum Intensity
Projection (Sagittal or
Coronal)





Alpha Fusion



Adjust aspect ratio
with Slice Thickness 5:1

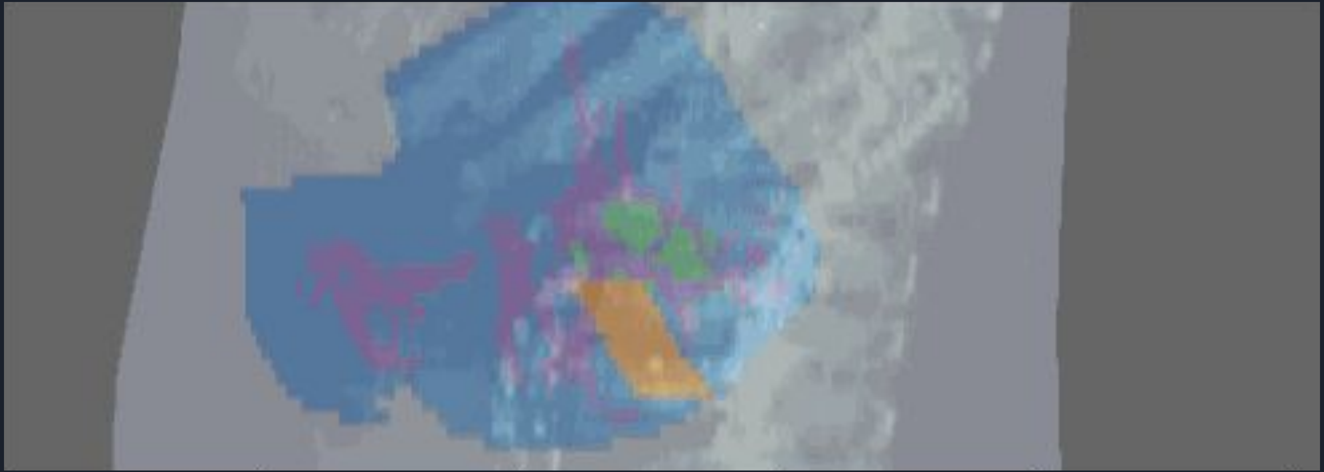
New Frame

CT img: **Bone**

Segmentation: **Set1**

1.5 GIF animation

Final Result

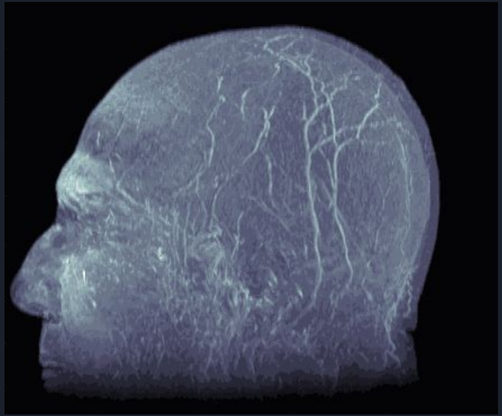


2.

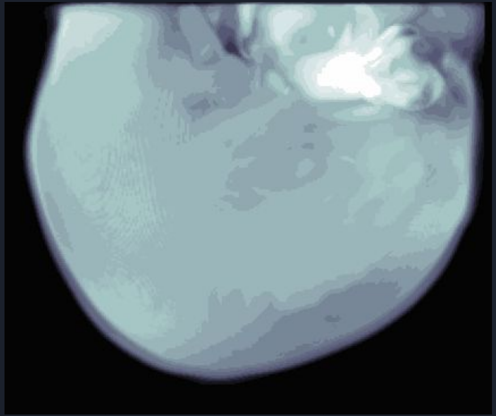
3D Rigid Coregistration

2.1 DICOM Files

We rearrange the brain slices



We Inspect Phantom and Atlas brains



2.2 Match Dimensions Physically

	Pixel Spacing [row,col]
Patient's Brain	[0.5078,0.5078]
Phantom Brain	[1,1]

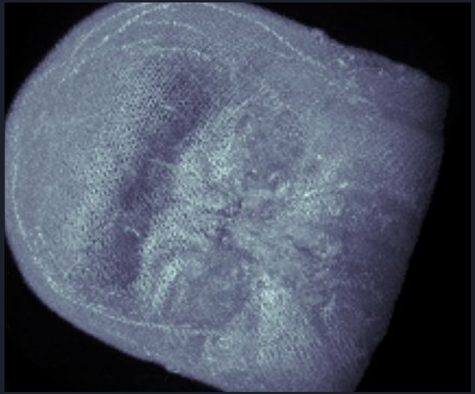
By a factor of 0.5078

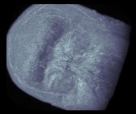
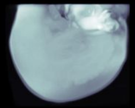
	Original Sizes	**Crop Phantom**	**Downsize Pt's Brain**	**Crop Pt's Brain**
Pt's Brain	(212, 512, 512)	(212, 512, 512)	(212, 259, 259)	(181, 217, 181)
Ph Brain	(193, 229, 193)	(181, 217, 181)	(181, 217, 181)	(181, 217, 181)
Atlas	(181, 217, 181)	(181, 217, 181)	(181, 217, 181)	(181, 217, 181)

2.3 Main Idea

Optimize a function that:

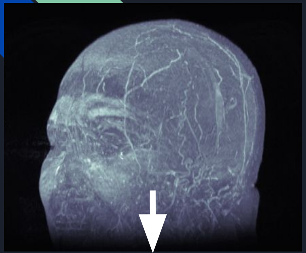
1. Transforms the Input Patient's Brain
2. Compute the difference/similarity with the phantom



$\mathrm{MSE} = \frac{1}{w \cdot h \cdot d} \sum \big($  $-$  $\big)^2$

Translation and then axial rotation

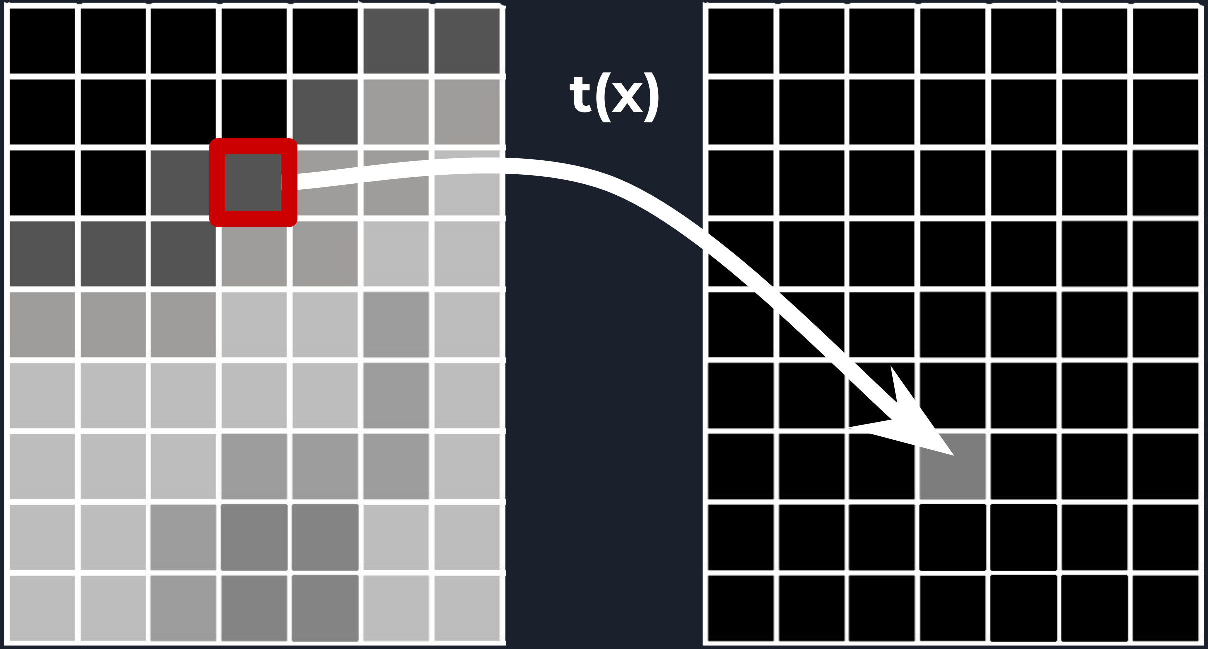
2.3 Transformations with Quaternions



[0,0,0]
[0,0,1]
...
[180,216,180]

Translation and then axial rotation: t(x)

[52,56,78]
[85,85,46]
...
[-55,350,67]



Input Patient’s Brain

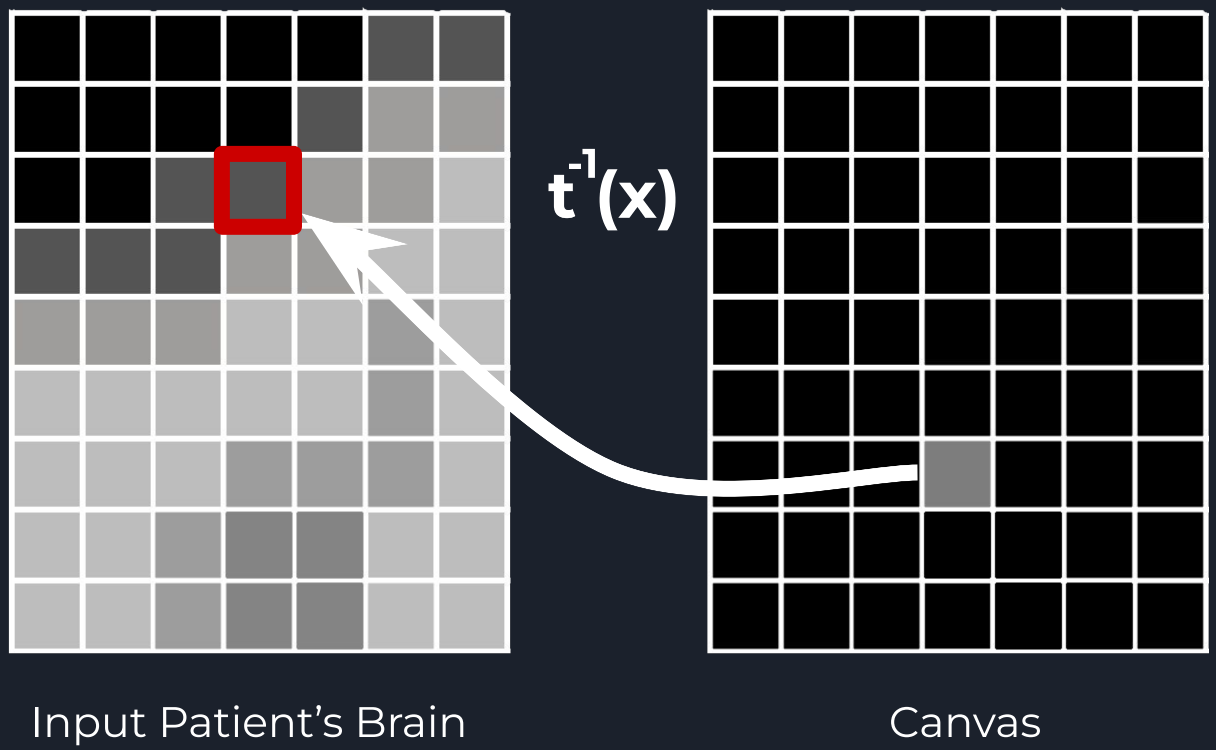
Canvas

2.3 Transformations with Quaternions

Problem: Unrepresented Pixels

2.3 Transformations with Quaternions

Solution: Inverse transformation approach



2.3 Transformations with Quaternions

Problem: For loop solution too slow → 7 109 137 pixels

Solution: Use numpy-quaternions

NumPy ndarray with dtype=quaternion

```
array([quaternion(0, 180, 216, 81),
       quaternion(0, 180, 216, 83),
       quaternion(0, 180, 216, 85),
       quaternion(0, 180, 216, 87),
       quaternion(0, 180, 216, 89),
       quaternion(0, 180, 216, 91),
       quaternion(0, 180, 216, 93),
       quaternion(0, 180, 216, 95),
       quaternion(0, 180, 216, 97),
       quaternion(0, 180, 216, 99),
       quaternion(0, 180, 216, 101),
       quaternion(0, 180, 216, 103),
       quaternion(0, 180, 216, 105),
       quaternion(0, 180, 216, 107),
       quaternion(0, 180, 216, 109),
```

Apply the transformation as in the lectures

$$\alpha \in [0, 2\pi),\ \ \vec{v} \in \mathbb{R}^3 \text{ unitary,}$$
$$\vec{p} \in \mathbb{R}^3,\ \ \vec{p}' = \mathrm{AxialRot}_{\vec{v},\alpha}(p),$$
$$q = \cos(\alpha/2) + \sin(\alpha/2) \cdot v,$$
$$\implies p' = q \cdot p \cdot q^*$$

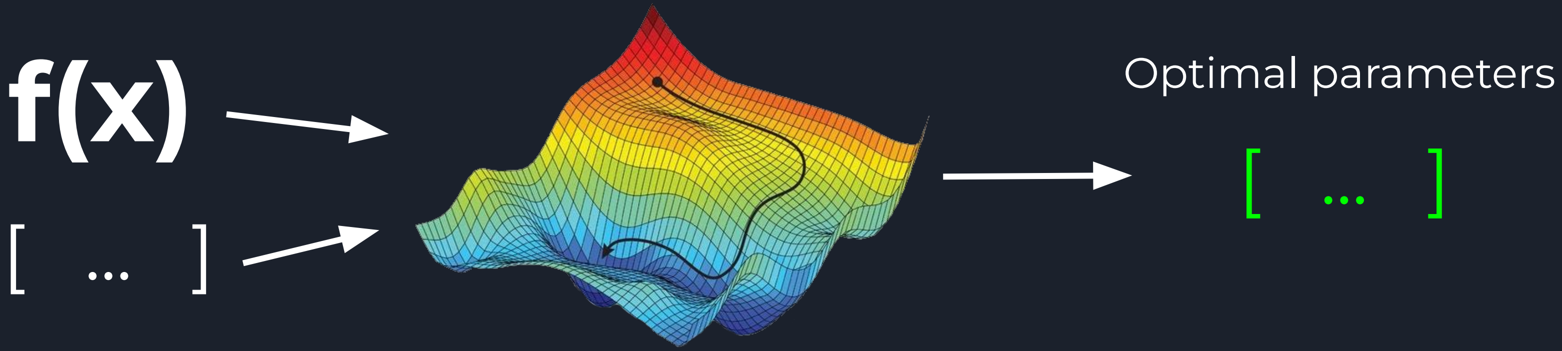
```
q_star = np.quaternion.conjugate(q_ax_rot)
q_tmp = q_indices * q_star
q_prime = q_ax_rot * q_tmp
```

2.4 Optimization method

We use scipy.optimize.minimize

Nelder-Mead method

Method	L-BFGS-B	BFGS	**Nelder-Mead**	Powell	CG	SLSQP	COBYLA	TNC
MSE	0.064	0.064	**0.058**	0.11	0.064	0.114	0.061	0.063
Time (s)	147	798	332	302	575	28	141	97

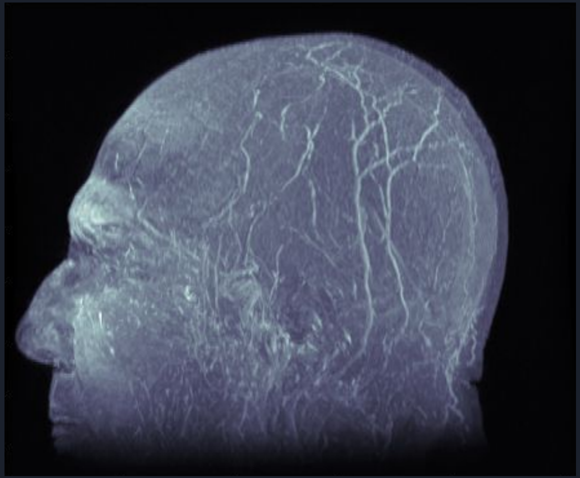


2.4 Optimization method

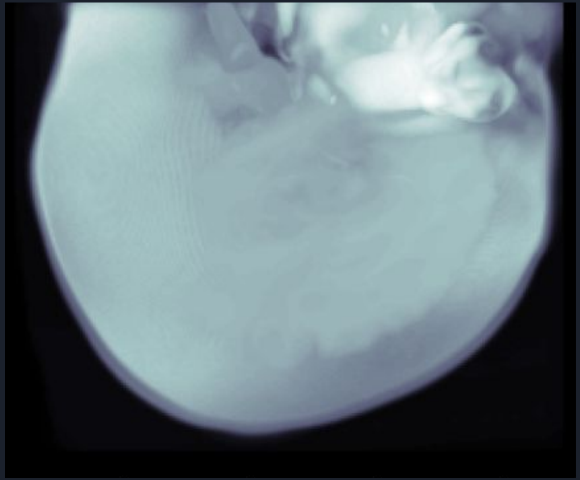
Initialization parameters

Problem: Input and reference brain are not in the same orientation

Input brain



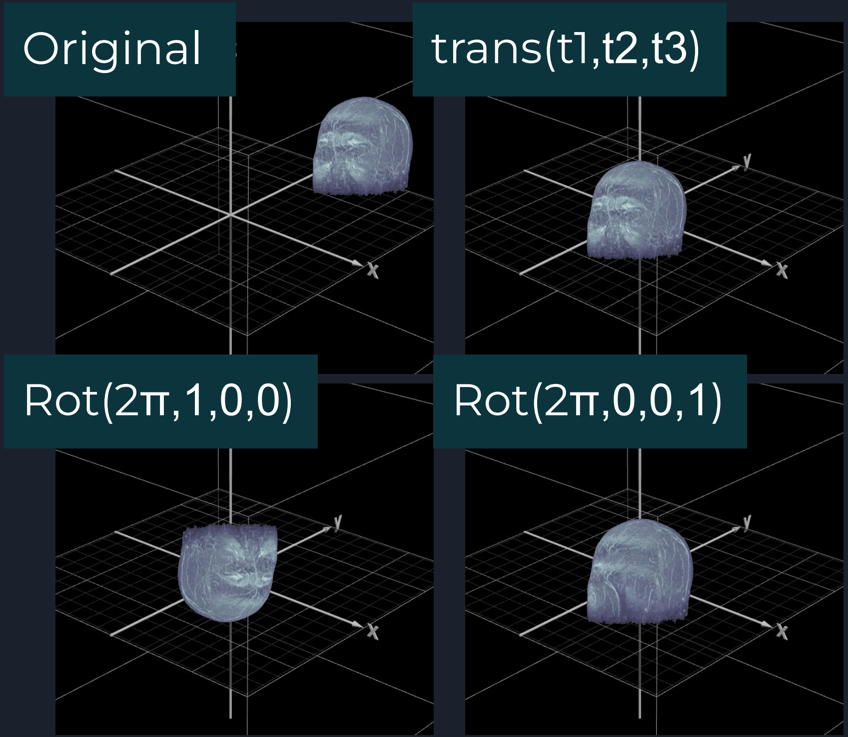
Reference brain

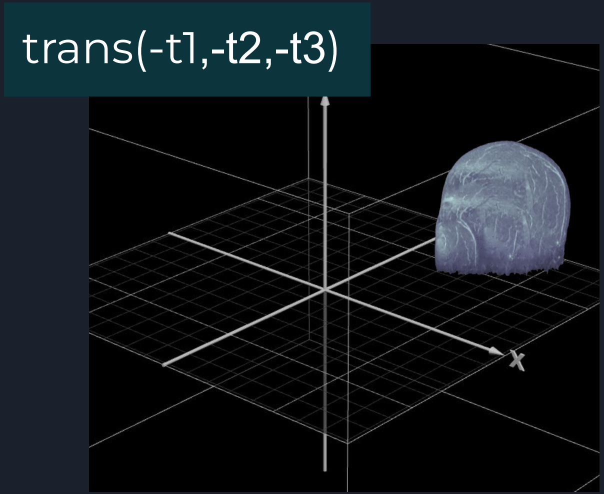


2.4 Optimization method

Initialization parameters

Solution: Initialize parameters to make them start in the same orientation

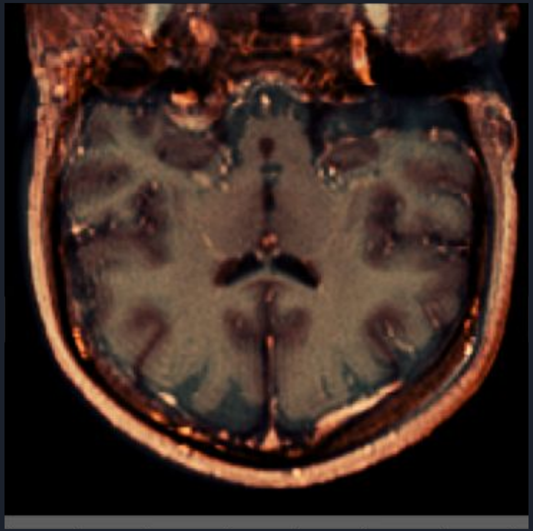


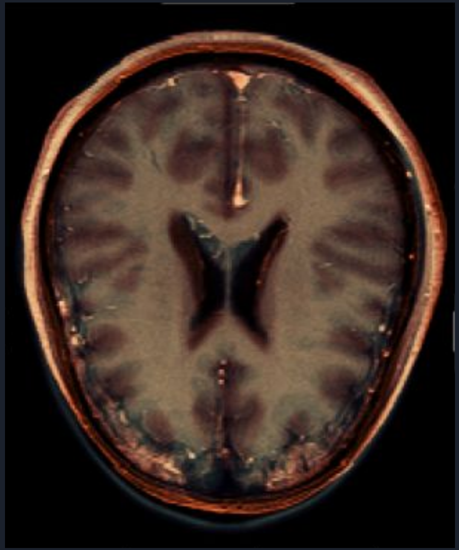


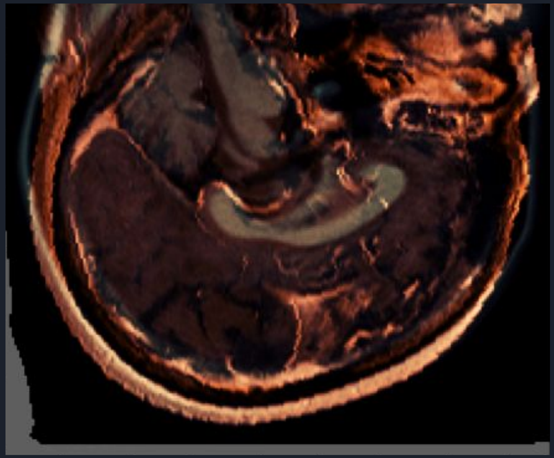
[-181,-217,0,π,0,0,-1]

2.5 Coregistration Results

Optimal parameters: [-181.22,-203.38,0,2.97,0,0,-1.06]







2.6 Thalamus Region

Mask extraction

- We extract all the Thalamus IDs from *AAL3_1mm.txt*
- We create a unified mask with all the pixels whose value belongs to any Thalamus ID

2.7 Thalamus Region

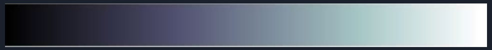
Transform to patient's input space

- We simply apply the inverse transformation to the thalamus mask.
- We apply alpha fusion to the mask and input brain.

Axial rotation and then translation

[181.22,203.38,0,2.97,0,0,1.06]

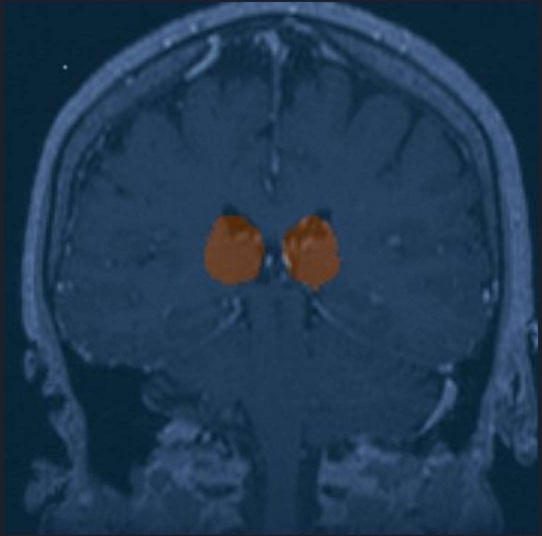
Input: **Bone**

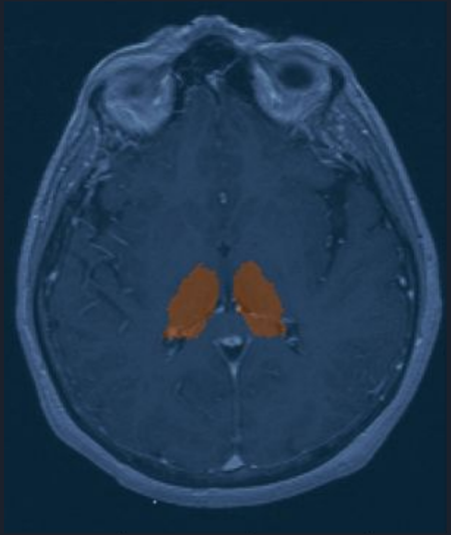


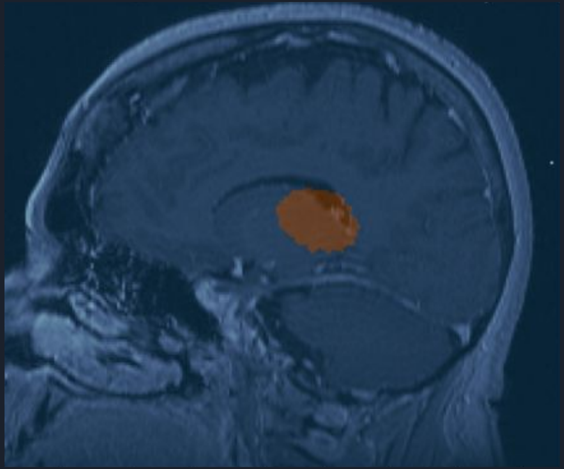
Thalamus mask: **tab10**

2.7 Thalamus Region

Final Result

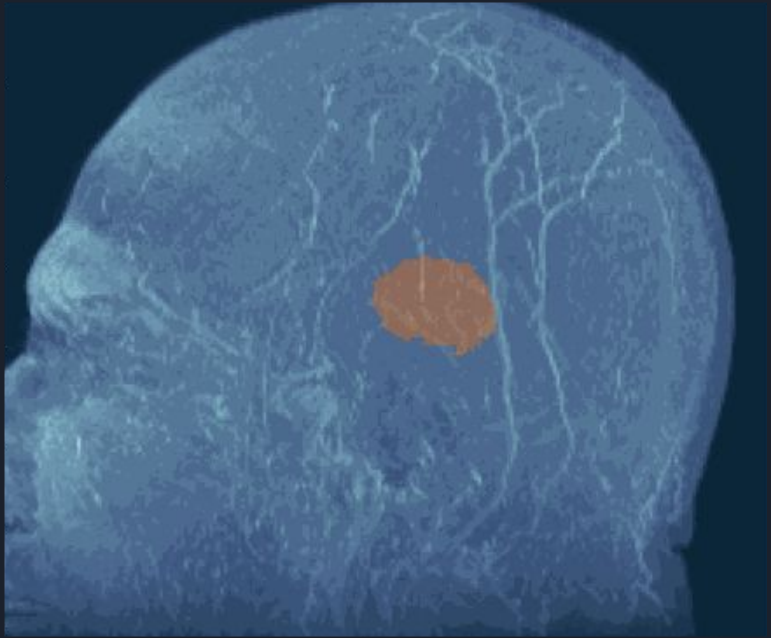






2.7 Thalamus Region

Final Result



Conclusions

- Inverse transformation + quaternions approach
- Best method for our problem: Nelder-Mead
- Hardcoded and specific data tailored aspects.
- Non suitable for real-time applications.
- Better visualization.

Thank you

Guillem Bibiloni Femenias