Volumes & Volumetric Timeseries

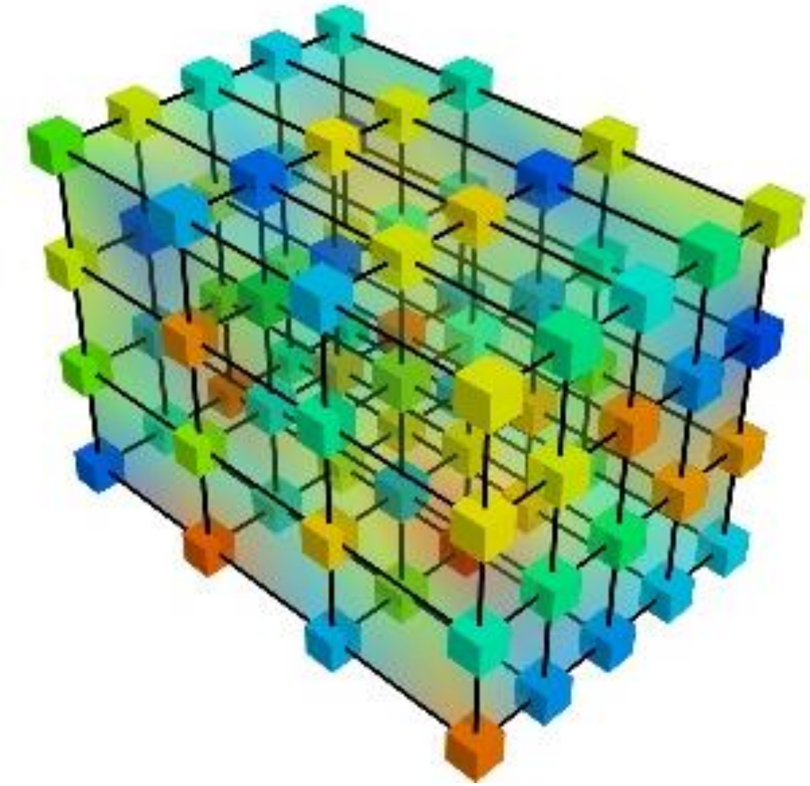
10.16.24

Learning Objectives

- Volumetric Viewing
- Volumetric Registration
- Surface detection/closure
- Masking

Volumetric Data

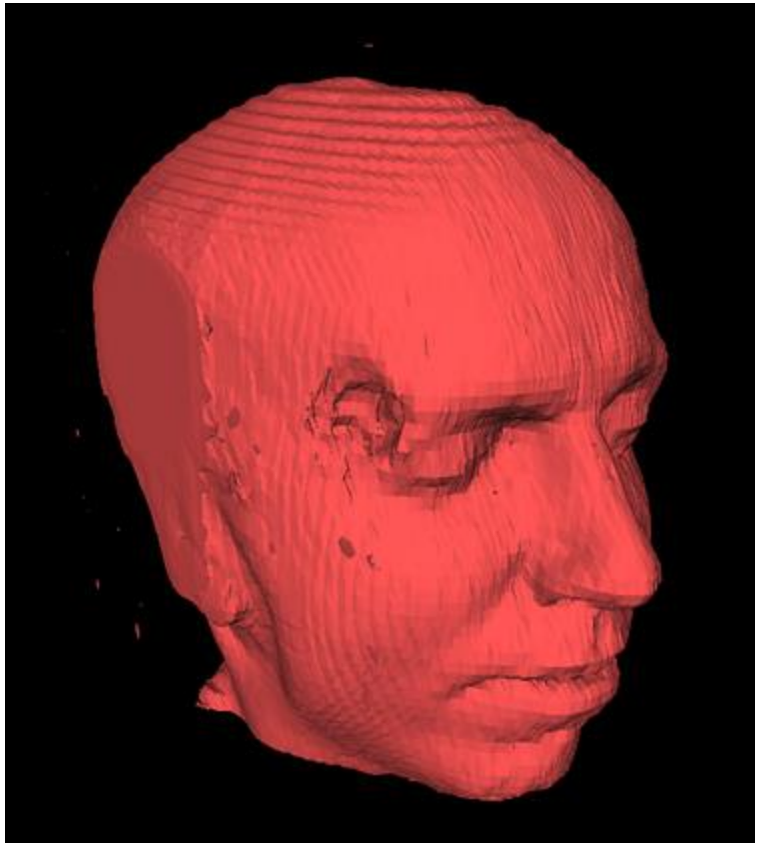
- Full reconstruction of a physical scene as a 3d array (X,Y,Z)
- Volumetric timeseries=4d: X, Y, Z, T
- Voxel=volumetric pixel
- More separable than regular images because we directly capture the empty-space (air) around objects



Volumetric Visualization

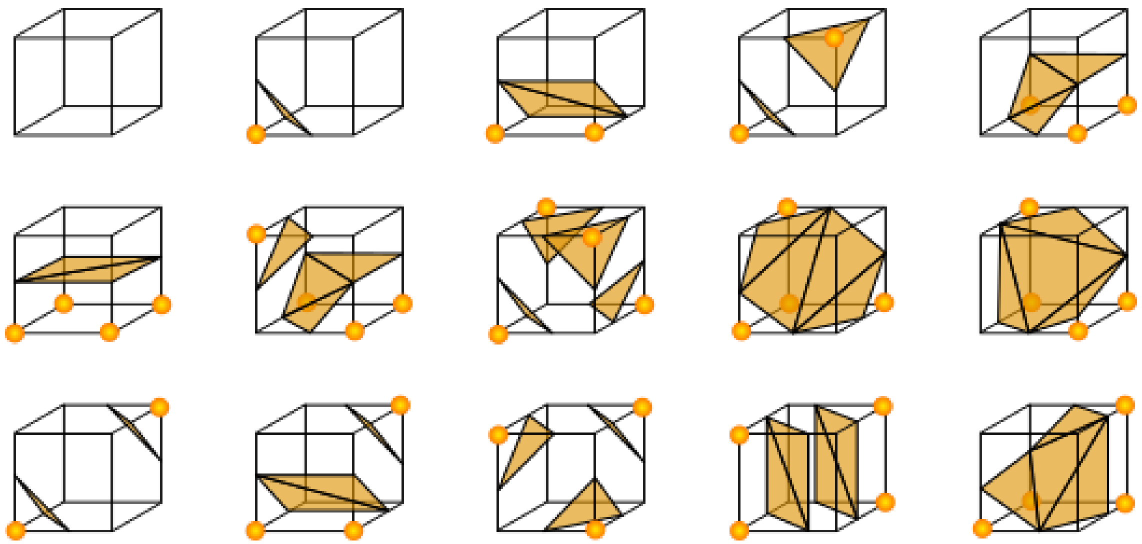
- Graphics are composed of meshes
- Mesh geometry is determined by **isocontours:** connecting parts of the volume with similar values
- Lighting effects provide important information

Mesh created by connecting voxels with ~ same values



Marching Cubes Algorithm

- Isosurface calculation
- Generates meshes to connect neighboring voxels with the same value (2x2x2 grid)
- Meshes additively extended as neighborhood window moved through the volume

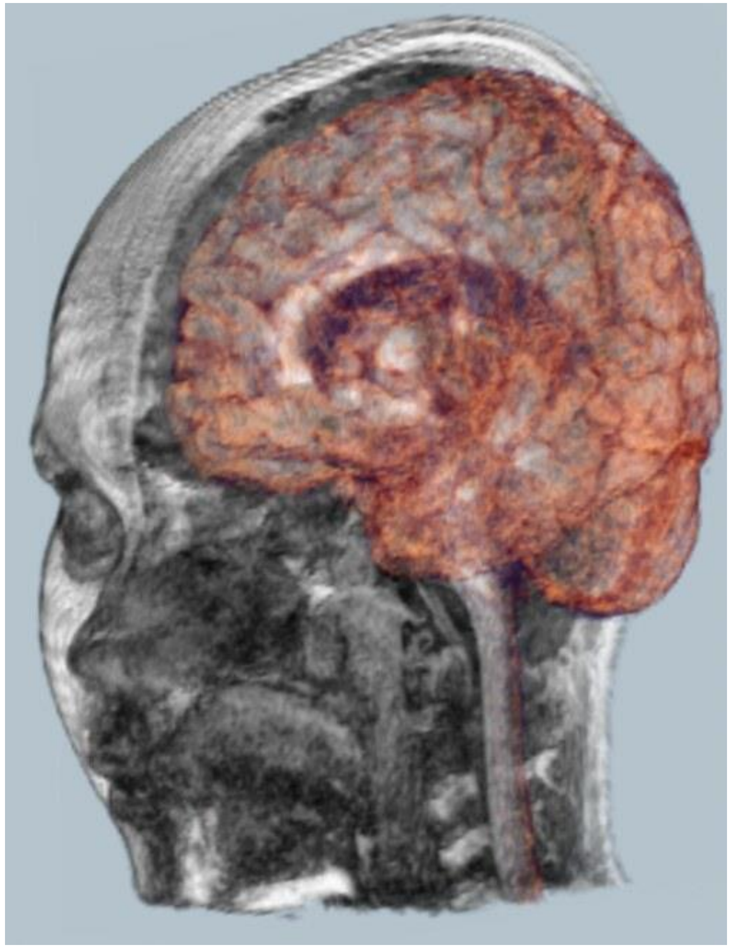


Volumetric Visualization

- Layered structures through transparent meshes

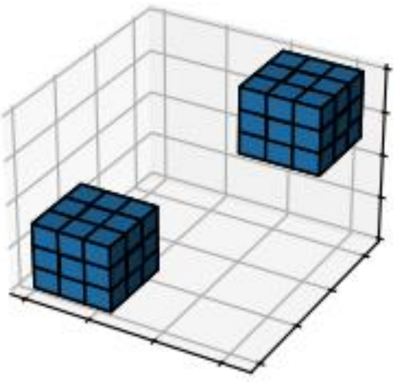
- Unlike images, color-data (when available) is specified separately.

- You have to specify one volume (X,Y,Z) to build isosurface meshes and then can use the full data (X,Y,Z,color) to color the meshes

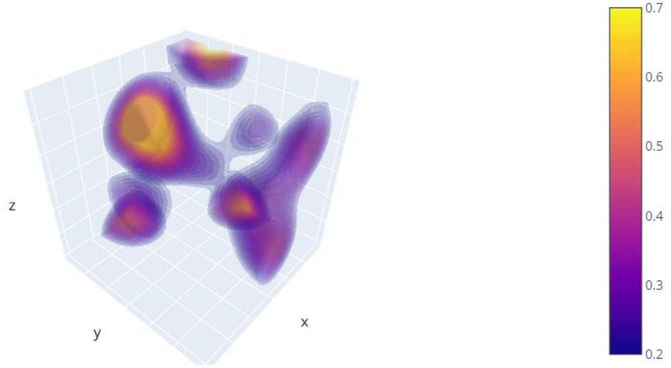


Visualization Modules

- Matplotlib: uses CPU only
  - voxels plot: cube-faithful, assumes isotropic
  - after mesh conversion: plot_trisurf
  - poor performance



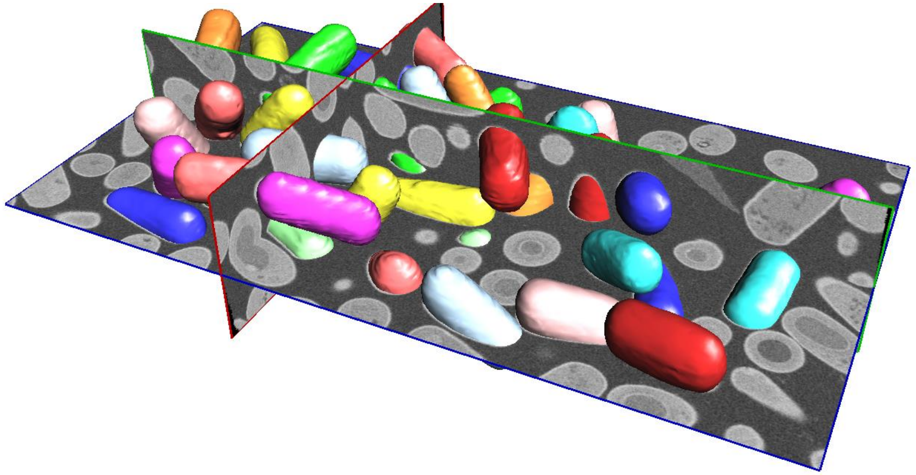
- Plotly volume: browser-based renderer
  - Automatic isosurface generation for multiple levels



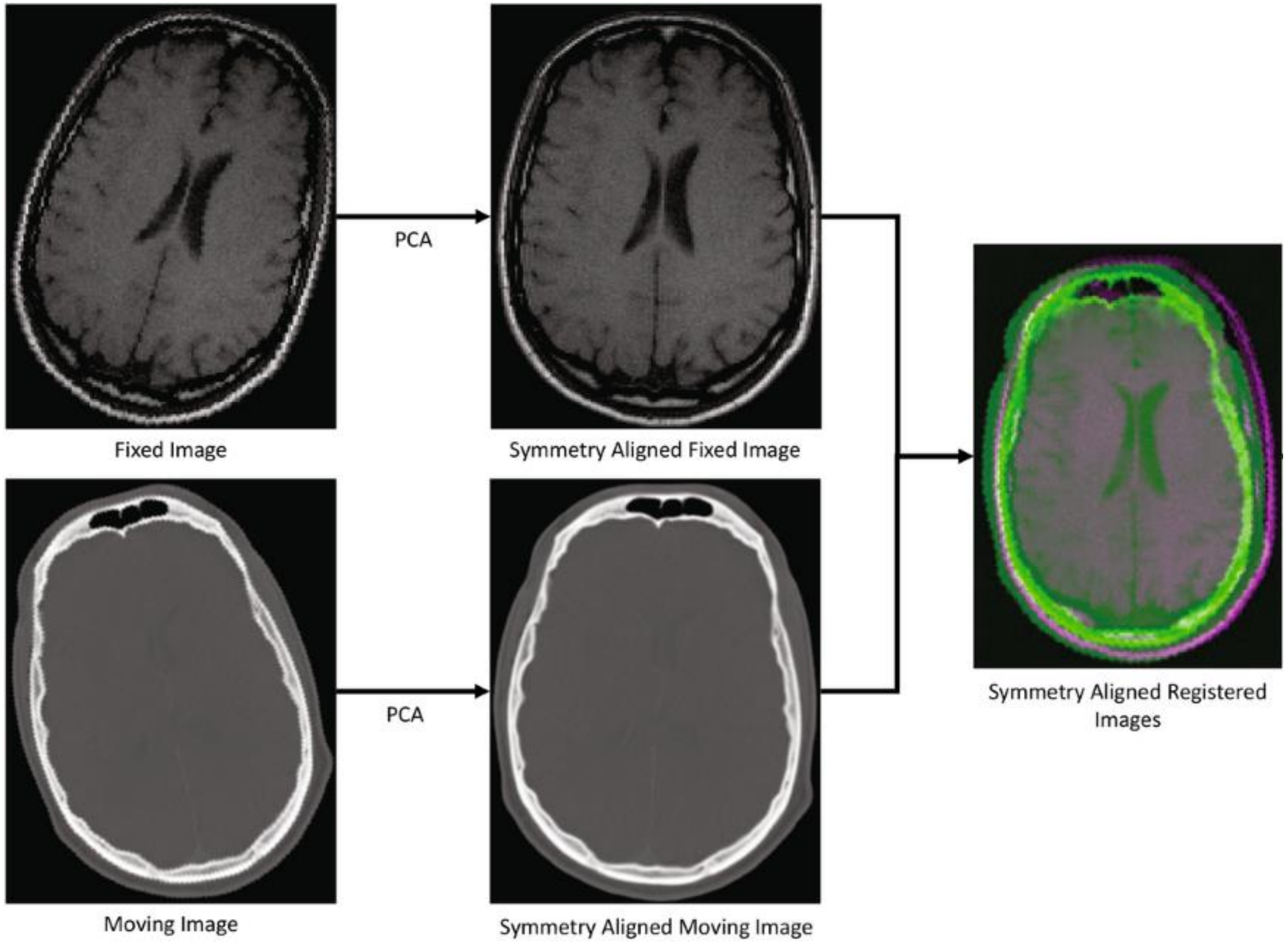
Visualization Modules

- Napari (built on Qt and visPy for GPU)
  - Interactive GUI integrated with Python:

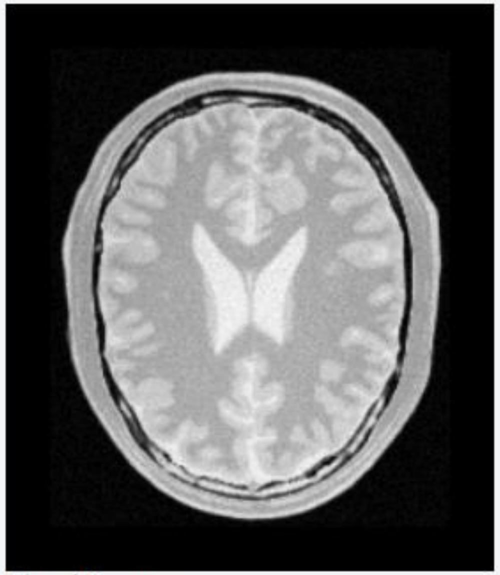
- Simple ITK+ FIJI/ImageJ
  - Python volume export with sitk.Show() to ImageJ GUI
  - FIJI/ImageJ is multilanguage



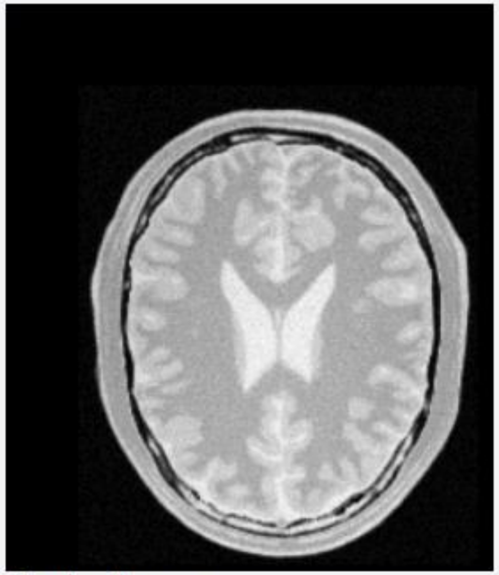
Registration



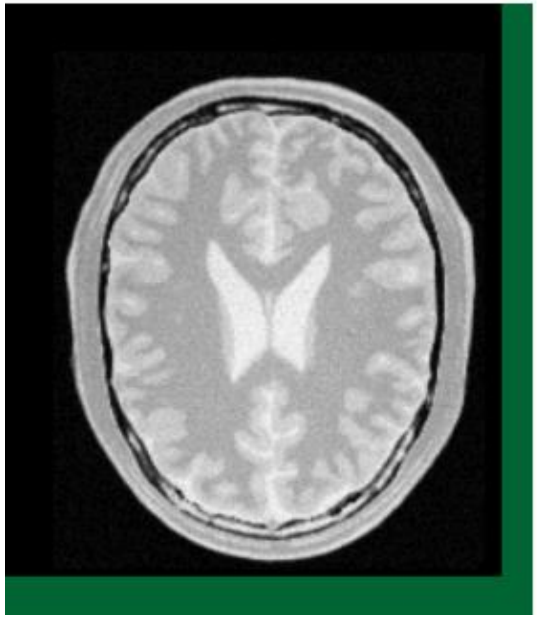
Registration



Fixed Image

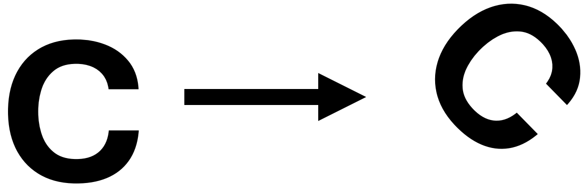


Moving Image

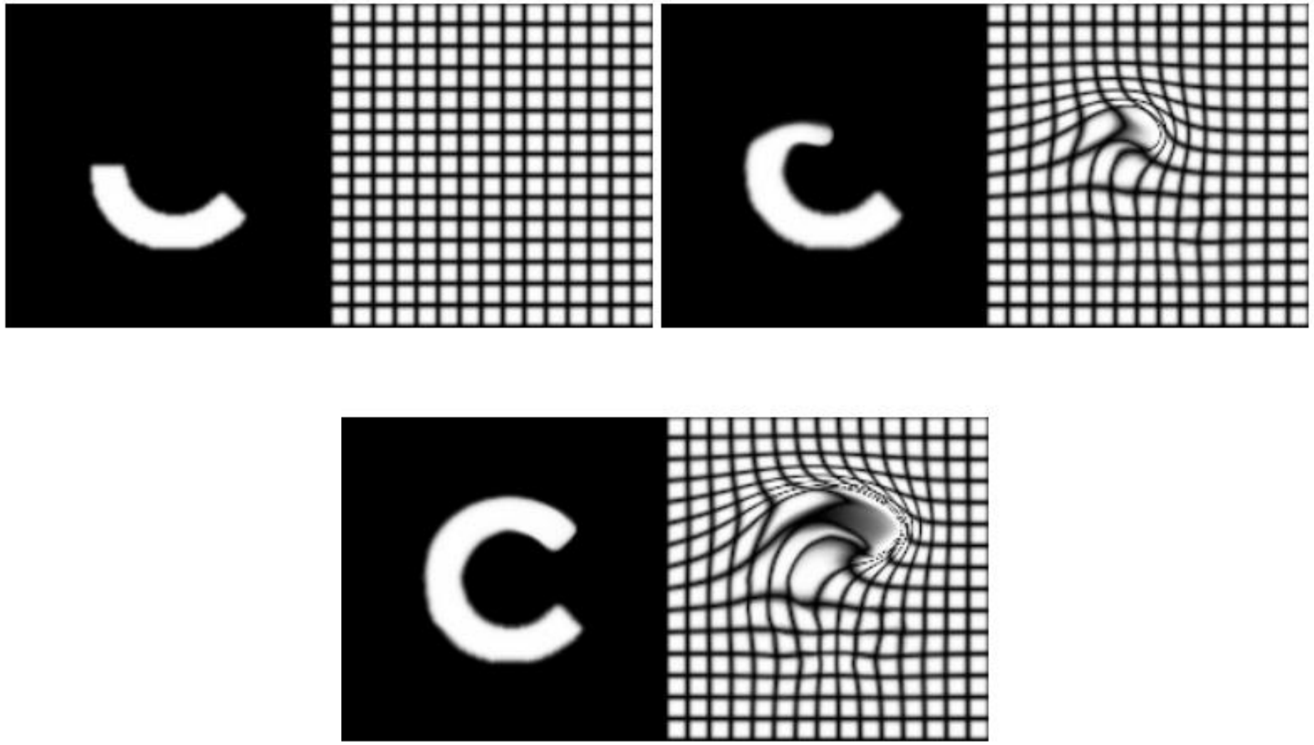


Affine vs. Nonlinear Registration

Affine



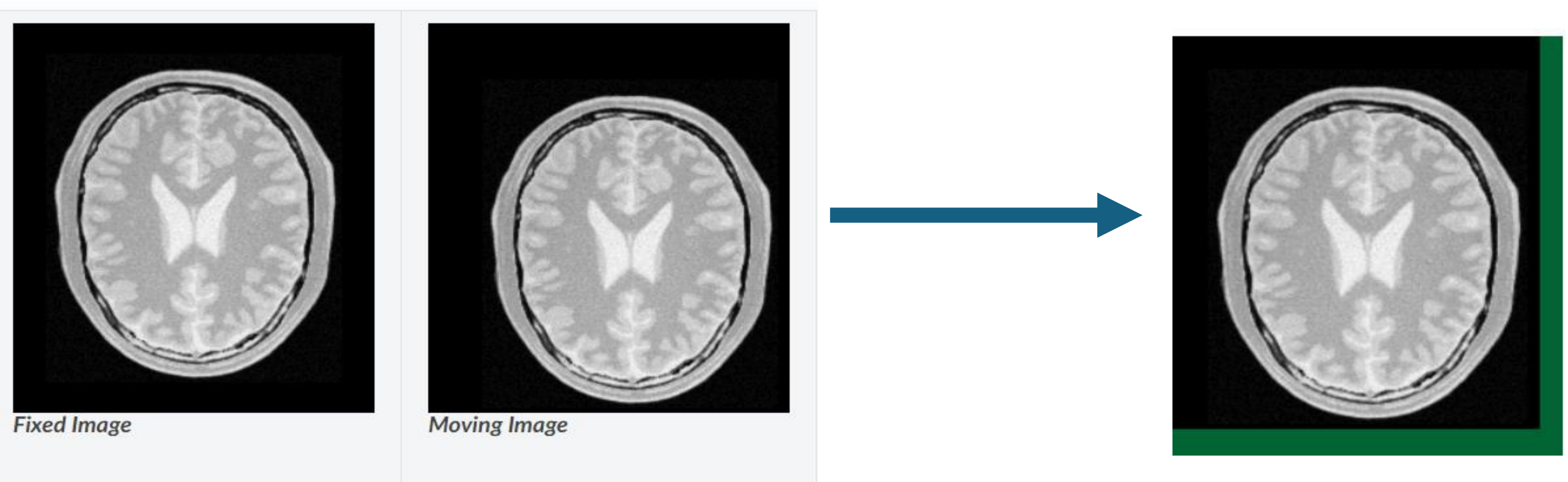
Non-Euclidean



Registration

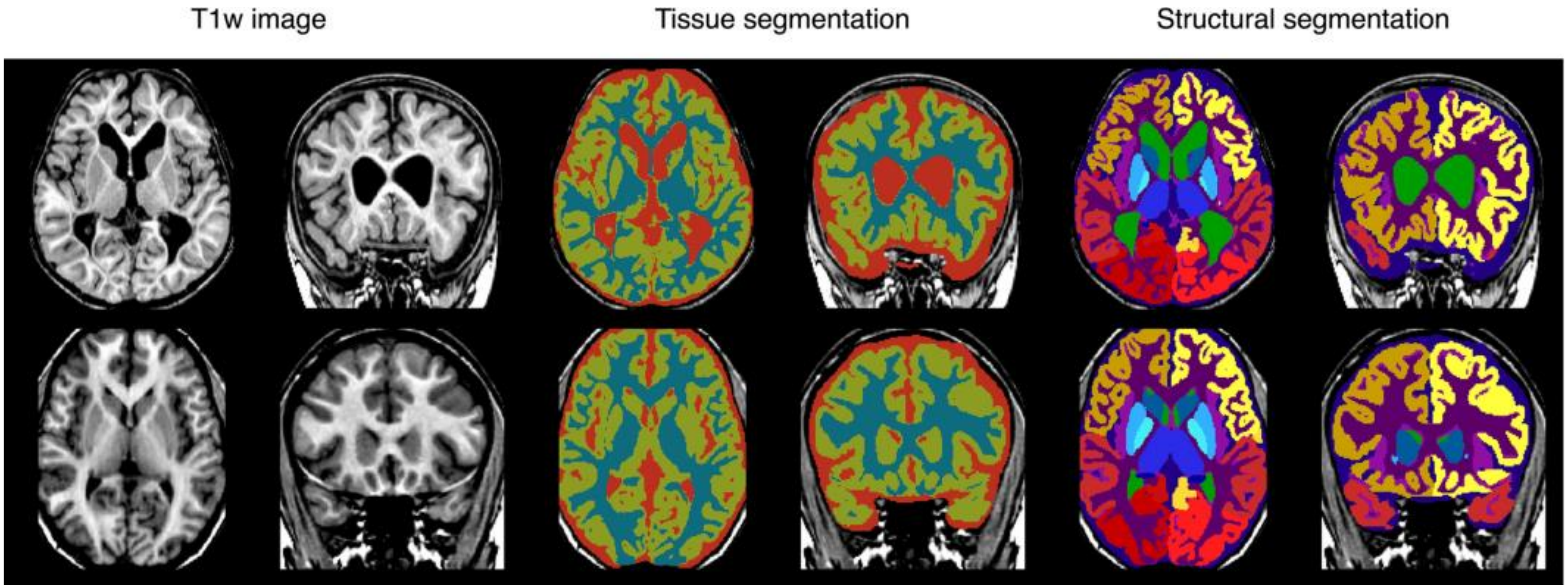
- Often a first approximation is ideal before starting heavy machinery

- Ideas for rough alignment?



Onto demos

Segmentation



Sobel Gradient Approximation

- Edges are where the change in intensity (gradient) is large
- Sobel kernels: gradient in one direction, smoother in the other:

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, \qquad G_y = G_x^T$$

- Full Sobel is gradient magnitude: $G_x^2 + G_y^2$

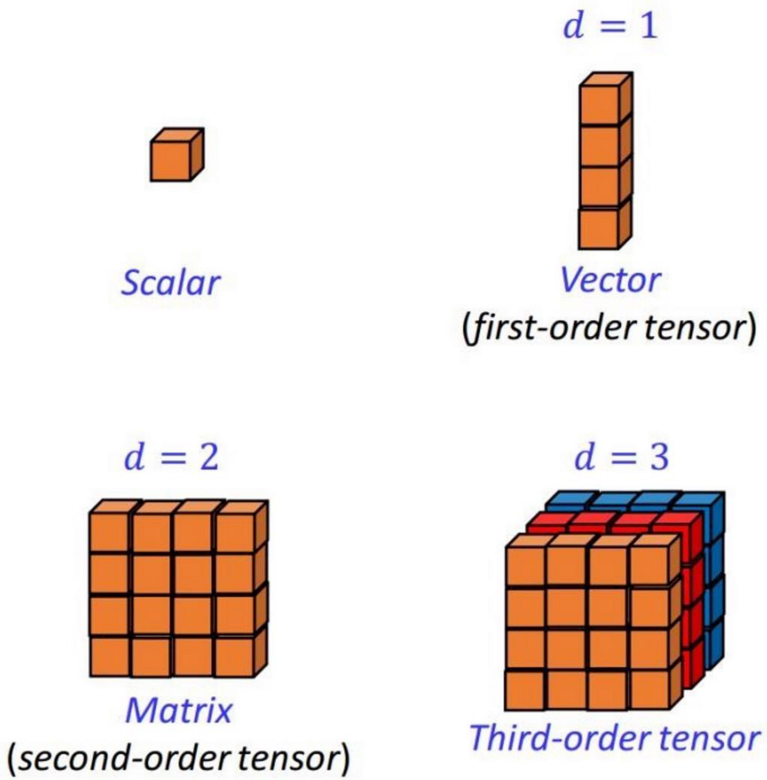
Edge Detection in 3d

Sobel kernel (2d)

$$G_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}$$

Sobel tensor (3d)

$$\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}^{(1)} \otimes \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}^{(2)} \otimes \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}^{(3)}$$



Onto demos

Fin