Medical image Processing and visualization for Alder Hey hospital

Supervisor: Roberto ,Dr Ferrero. Assessor: Spencer, Prof Joseph
Student: Ziwei Tao

UNIVERSITY OF LIVERPOOL

• Background Information

In the modern society, as the rapid development of the diagnostic technique, such as Magnetic Resonance Imaging (MRI), a 3D image could be stored as a sequence of 2D slices and these 2D images could also be processed by different software to create a 3D visualization. However, these software processing images are based on different coordinate systems

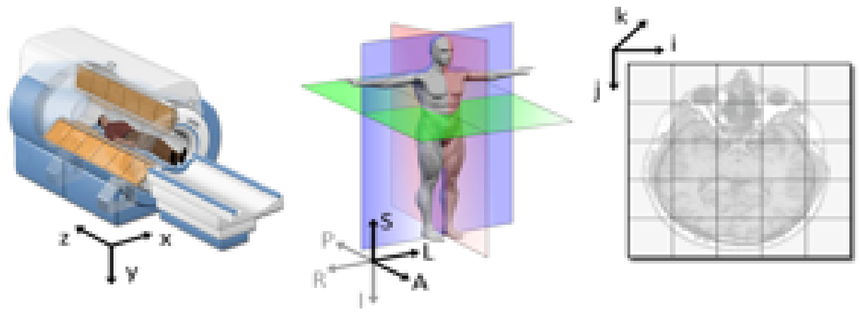
• Objectives

This project aims to develop a software interface which could support the clinician's work by easily transforming 2D images in different coordinate systems, according to the clinical needs.

1. Simulate DICOM dataset on an open source software platform.
2. Find a mathematical model for coordinate transformation.
3. Create a software interface on Python to achieve coordinate transformation and present 2D medical images.

• Theory

1. Three coordinate systems

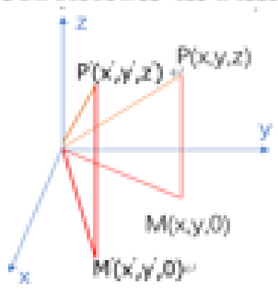


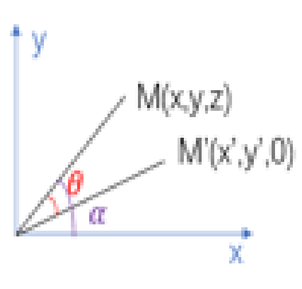
World coordinate system, Anatomical coordinate system, Image coordinate system.

2. Coordinate transformation

Rotation matrix:

1. Rotation around z-axis





$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

$$R_z(\theta) = \begin{bmatrix} cos\theta & -sin\theta & 0 \\ sin\theta & cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2. Rotation matrix in 3D dimensions

$$v' = Rv$$

$$R = \begin{bmatrix} \cos(x',x) & \cos(y',x) & \cos(z',x) \\ \cos(x',y) & \cos(y',y) & \cos(z',y) \\ \cos(x',z) & \cos(y',z) & \cos(z',z) \end{bmatrix} \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(x',x) & \cos(y',x) & \cos(z',x) \\ \cos(x',y) & \cos(y',y) & \cos(z',y) \\ \cos(x',z) & \cos(y',z) & \cos(z',z) \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

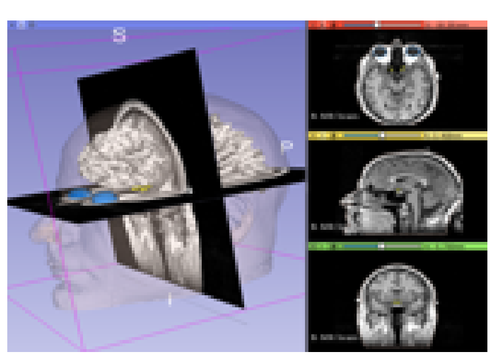
V' (x', y', z') is the point corresponding to v=(x, y, z) in a transformed coordinate system and R is the rotation matrix

Transformation matrix:

$$v' = Qv \qquad Q = \begin{bmatrix} \cos(x',x) & \cos(x',y) & \cos(x',z) \\ \cos(y',x) & \cos(y',y) & \cos(y',z) \\ \cos(z',x) & \cos(z',y) & \cos(z',z) \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} \cos(x',x) & \cos(x',y) & \cos(x',z) \\ \cos(y',x) & \cos(y',y) & \cos(y',z) \\ \cos(z',x) & \cos(z',y) & \cos(z',z) \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \end{bmatrix}.$$

3. 3D Model Segmentation



• Methods

1. Find the origin of Scanner Coordinate System.

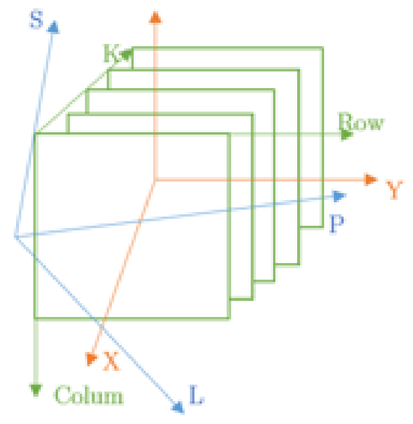
Assume that point C is the center of the 3D image,. Point A is (a1, a2, a3), point B is (b1, b2, b2) and O (x, y, z) is the origin of the middle slice

$$|\overrightarrow{OA}| = (a1 - x, a2 - y, a3 - z)$$
$$|\overrightarrow{OB}| = (b1 - x, b2 - y, b3 - z)$$
$$|\overrightarrow{OA}| = \sqrt{(a1-x)^2 + (a2-y)^2 + (a3-z)^2} = 240mm$$
$$\cos\theta_{rx} = \frac{\overrightarrow{OA}\cdot\overrightarrow{n_x}}{|\overrightarrow{OA}|\cdot|\overrightarrow{n_x}|} = \frac{a_1 - x}{240mm \cdot 1}$$

C = ((a1+b1/2), (a2+b2/2), (a3+b3/2))

$$= (120(\cos\theta_{rx} + \cos\theta_{cx}) + x,\ 120(\cos\theta_{ry} + \cos\theta_{cy}) + y, 120(\cos\theta_{rz} + \cos\theta_{cz}) + z)$$
$$\approx (0.373317, -13.534750, -0.774501)$$

2. Find the Transformation Equation



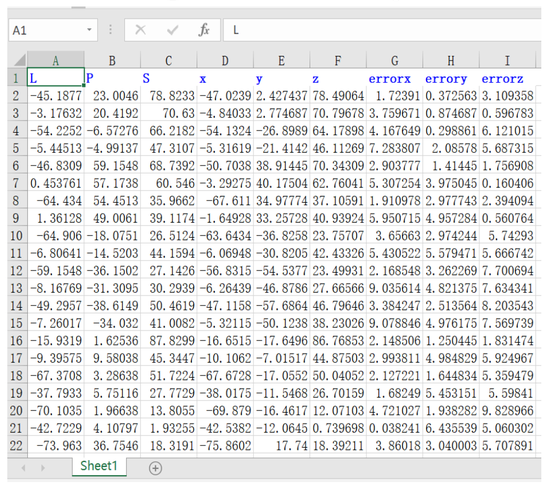
$$(x, y, z) \rightarrow (-k, row, -column)$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} \cos\theta_{xL} & \cos\theta_{yL} & \cos\theta_{zL} \\ \cos\theta_{xp} & \cos\theta_{yp} & \cos\theta_{zp} \\ \cos\theta_{xs} & \cos\theta_{ys} & \cos\theta_{zs} \end{bmatrix} \times \begin{bmatrix} L \\ P \\ S \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

$$\rightarrow \begin{bmatrix} -k \\ r \\ -c \end{bmatrix} = \begin{bmatrix} -\cos\theta_{kL} & \cos\theta_{rL} & -\cos\theta_{cL} \\ -\cos\theta_{kp} & \cos\theta_{rp} & -\cos\theta_{cp} \\ -\cos\theta_{ks} & \cos\theta_{rs} & -\cos\theta_{cs} \end{bmatrix} \times \begin{bmatrix} L \\ P \\ S \end{bmatrix} + \begin{bmatrix} C_1 \\ C_2 \\ C_3 \end{bmatrix}$$

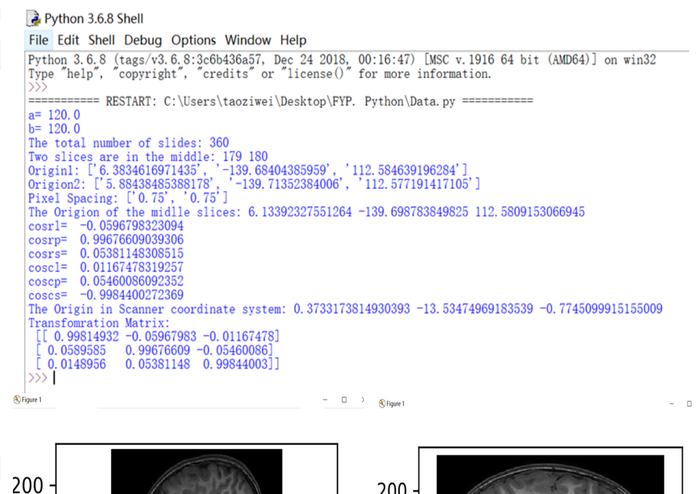
.Testing Results

1. Outputs in Python. 2. Transformed coordinates



	A	B	C	D	E	F	G	H	I
1	L	P	S	x	y	z	errorx	errory	errorz
2	-45.1877	23.0046	78.8233	-47.0239	2.427437	78.49064	1.72391	0.372563	3.109358
3	-3.17632	20.4192	70.63	-4.84033	2.774687	70.79678	3.759671	0.874687	0.596783
4	-54.2252	-6.57276	66.2182	-54.1324	-26.8989	64.17898	4.167649	0.298861	6.121015
5	-5.44513	-4.99137	47.3107	-5.31619	-21.4142	46.11269	7.283807	2.08578	5.687315
6	-46.8309	59.1548	68.7392	-50.7038	38.91445	70.34309	2.903777	1.41445	1.756908
7	0.453761	57.1738	60.546	-3.29275	40.17504	62.76041	5.307254	3.975045	0.160406
8	-64.434	54.4513	35.9662	-67.611	34.97774	37.10591	1.910978	2.977743	2.394094
9	1.36128	49.0061	39.1174	-1.64928	33.25728	40.93924	5.950715	4.957284	0.560764
10	-64.906	-18.0751	26.5124	-63.6434	-36.8258	23.75707	3.65663	2.974244	5.74293
11	-6.80641	-14.5203	44.1594	-6.06948	-30.8205	42.43326	5.430522	5.579471	5.666742
12	-59.1548	-36.1502	27.1426	-56.8315	-54.5377	23.49931	2.168548	3.262269	7.700694
13	-8.16769	-31.3095	30.2939	-6.26439	-46.8786	27.66566	9.035614	4.821375	7.634341
14	-49.2957	-38.6149	50.4619	-47.1158	-57.6864	46.79646	3.384247	2.513564	8.203543
15	-7.26017	-34.032	41.0082	-5.32115	-50.1238	38.23026	9.078846	4.976175	7.569739
16	-15.9319	1.62536	87.8299	-16.6515	-17.6496	86.76853	2.148506	1.250445	1.831474
17	-9.39575	9.58038	45.3447	-10.1062	-7.01517	44.87503	2.993811	4.984829	5.924967
18	-67.3708	3.28638	51.7224	-67.6728	-17.0552	50.04052	2.127221	1.644834	5.359479
19	-37.7933	5.75116	27.7729	-38.0175	-11.5468	26.70159	1.68249	5.453151	5.59841
20	-70.1035	1.96638	13.8055	-69.879	-16.4617	12.07103	4.721027	1.938282	9.828966
21	-42.7229	4.10797	1.93255	-42.5382	-12.0645	0.739698	0.038241	6.435539	5.060302
22	-73.963	36.7546	18.3191	-75.8602	17.74	18.39211	3.86018	3.040003	5.707891

Sheet1



3. 2D medical image

• Conclusions

The main original contribution in this project is to build and apply a mathematical model to achieve coordinate transformation between two different coordinate systems. According to the testing results, although there are still some errors created during transformation, it could be seen that this mathematical model is correct and the output of the programming has a high level of similarity when comparing with the samples

• Future work

1. Improve the accuracy of the mathematical model and speed of coordinate transformation
2. Build a 3D image model directly by Python.
3. GPU applied in medical imaging can be improved to accelerate medical image processing