International Journal of Electrical and Electronics Engineers ISSN- 2321-2055 (E)
http://www.arresearchpublication.com IJEEE, Volume 07, Issue 01, Jan- June 2015

IMPLEMENTATION OF MULTIPLANAR RECONSTRUCTION IN CollabDDS USING JAVA TECHNIQUES

Gokul.R[1], J.Raamkumar[2], Mridu Pobon Rajkhowa[3]

[1]PG Scholar, SKP Engineering College, Tiruvannamalai, Tamil Nadu (India)
[2]Lecturer, Govt. Polytecnic College, Tiruvannamalai, Tamil Nadu (India)
[3]Scientist-B, NIC, New Delhi (India)

ABSTRACT

We propose to implement the conversion of slice views from DICOM images (CT Scan & MRI data) using java coding and slice orientation (direction cosine of the row and column with respect to the slice view of the image). Pixel size and Pixel Spacing are calculated with the help of row and column array from DICOM data for generating MPR view. For MPR different Affine Transformation like Scaling, Reflection, Rotation, Shear, Translation etc are to be applied on the image pixels.

Keywords: Affine Transformassion, Collab DDS, DICOM, MPR

I. INTRODUCTION

1.1 About CollabDDS

CollabDDS provides an integrated online environment to visualise medical and dental images for diagnosis and treatment planning. For visualization, different formats of data which can be loaded into CollabDDS like X-ray Images, JPEG/GIF/TIFF/WBMP/BMP /PNG, DICOM images (CT scan & MRI data), series. CollabDDS also provides excellent Image Manipulation features Remote Health Centres can be connected to expert radiologists and doctors in Centres of Excellence, by suitable tools and channel for data transmission and diagnosis. The repository of images with annotations by experts can also be used for education and research purposes. The high-bandwidth and low-latency capability of the National Knowledge Network (NKN) provides the ideal platform for real time collaboration using CollabDDS.

1.2 About MPR

In scientific visualization and computer graphics, Multiplanar Reformat technique is used in two-dimentional tomographic imaging (computed tomography). MPR allows images to be created from the original axial plane in coronal and sagittal plane and vice versa.

Multiplanar Reconstruction (MPR) allows images to be created from the original axial plane in either the coronal, sagittal, or oblique plane. Spiral/helical CT has greatly improved the quality of MPRs by eliminating the 'stair-step' artefacts that used to occur with conventional slice by slice CT. Multislice CT has further improved the quality of MPRs as it allows isotropic imaging in which the image quality of the reconstructed MPR is the same as the original axial image.

Multi-frame DICOM images contain a collection of slices of the body. For example, considering each axial slice represents an (X-Y) plane of the body, and then a collection of slices will represents Z axis.

In the simplest case, multi-planar reconstruction (MPR) involves generating perspectives at right angles to a stack of axial slices so that coronal and sagittal images can be generated. The described views can be graphically represented as the following:

- Axial view
- Sagittal view
- Coronal view

So if we have a multi-frame DICOM image that contains F frames of images with (W x H) dimension, then W would be the X axis, H would be the Y axis and F would be the Z axis.

The coronal view could be considered as the oblique view with 0 degree rotation angle about Z axis, while the sagittal view could be considered as the oblique view with 90 degree rotation angle about Z axis.

It is possible to perform MPR not only from multi-frame DICOM images, but also from a collection of single-frame DICOM images with the same width and height.

•**Axial**: The axial plane passes through the body from anterior to posterior and divides it into superior and inferior sections.

•**Coronal**: The coronal plane passes through the body from left to right and divides it into anterior to posterior sections.

•**Sagittal**: The sagittal plane passes through the body from anterior to posterior and divides it into left and right sections.

It is proposed to implement **Multiplanar Reformat** in **CollabDDS** as the reconstruction of the other two views would help the Radiology better visualize the images.

1.3 About DICOM

Digital Imaging and Communications in Medicine (DICOM) is a standard for handling, storing, printing, and transmitting information in medical imaging. It includes a file format definition and a network communications protocol. The communication protocol is an application protocol that uses TCP/IP to communicate between systems. DICOM files can be exchanged between two entities that are capable of receiving image and patient data in DICOM format. The National Electrical Manufacturers Association (NEMA) holds the copyright to this standard. It was developed by the DICOM Standards Committee, whose members are also partly members of NEMA.

DICOM enables the integration of scanners, servers, workstations, printers, and network hardware from multiple manufacturers into a picture archiving and communication system (PACS). The different devices come with DICOM conformance statements which clearly state which DICOM classes they support. DICOM has been widely adopted by hospitals and is making inroads in smaller applications like dentists' and doctors' offices.

1.3.1 Data format

DICOM differs from some, but not all, data formats in that it groups information into data sets. That means that a file of a chest x-ray image, for example, actually contains the patient ID within the file, so that the image can never be separated from this information by mistake. This is similar to the way that image formats such as JPEG can also have embedded tags to identify and otherwise describe the image.

A DICOM data object consists of a number of attributes, including items such as name, ID, etc., and also one special attribute containing the image pixel data (i.e. logically, the main object has no "header" as such: merely a list of attributes, including the pixel data). A single DICOM object can have only one attribute containing pixel data. For many modalities, this corresponds to a single image. But note that the attribute may contain multiple "frames", allowing storage of cine loops or other multi-frame data. Another example is NM data, where an NM image, by definition, is a multi-dimensional multi-frame image. In these cases, three- or four-dimensional data can be encapsulated in a single DICOM object. Pixel data can be compressed using a variety of standards, including JPEG, JPEG Lossless, JPEG 2000, and Run-Length encoding (RLE). LZW (zip) compression can be used for the whole data set (not just the pixel data), but this has rarely been implemented.

II. METHOD

2.1 DICOM image Parameters

Following the DICOM image Parameters are going to be used in MPR module.

2.1.1 Patient Orientation.

The Patient Orientation (0020, 0020) relative to the image plane shall be specified by two values that designate the anatomical direction of the positive row axis (left to right) and the positive column axis (top to bottom). The first entry is the direction of the rows, given by the direction of the last pixel in the first row from the first pixel in that row. The second entry is the direction of the columns, given by the direction of the last pixel in the first column from the first pixel in that column. Anatomical direction shall be designated by the capital letters: A (anterior), P (posterior), R (right), L (left), H (head), F (foot). Each value of the orientation attribute shall contain at least one of these characters. If refinements in the orientation descriptions are to be specified, then they shall be designated by one or two additional letters in each value. Within each value, the letters shall be ordered with the principal orientation designated in the first character."

2.1.2 Image Position

The Image Position (0020, 0032) specifies the x, y, and z coordinates of the upper left hand corner of the image; it is the center of the first voxel transmitted.

2.1.3 Image Orientation

Image Orientation (0020, 0037) specifies the direction cosines of the first row and the first column with respect to the patient.

2.2 MPR Module

Multi-planar reconstruction (MPR) module images are re-formats at arbitrary planes, defined by the operator, using the pixel data from a stack of planar images (base images).

The digital value for each pixel is assigned to a virtual voxel with the same thickness as the slice thickness of the base image. This yields a volume of data that represents the scanned object. The MPR module uses the virtual voxel data to create the pixel values for the reconstructed images. When the dimensions of the scanned voxels (as set by slice thickness and in-plane resolution) are equal, the data set is said to be isotropic. Where the dimensions of the scanned voxel are not equal in all three planes, the data set is said to be anisotropic. Isotropic data yields the best reconstructions.

The viewport is described by its origin, its row unit vector, column unit vector and a normal unit vector (derived from the row and column vectors by taking the cross product). Now if one moves the origin to 0,0,0 and rotates this viewing plane such that the row vector is in the +X direction, the column vector the +Y direction, and the normal in the +Z direction, then one has a situation where the X coordinate now represents a column offset in mm from the localizer's top left hand corner, and the Y coordinate now represents a row offset in mm from the localizer's top left hand corner, and the Z coordinate can be ignored. One can then convert the X and Y mm offsets into pixel offsets using the pixel spacing of the localizer image.

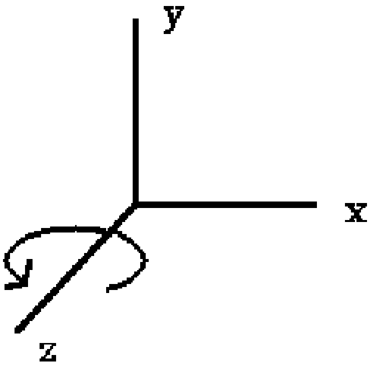
The actual rotations can be specified entirely using the direction cosines that are the row, column and normal unit vectors.

2.3 Module Calculations

2.3.1 3D Rotation

3D Rotation is more complicated than 2D rotation since we must specify an axis of rotation. In 2D the axis of rotation is always perpendicular to the xy plane, i.e., the Z axis, but in 3D the axis of rotation can have any spatial orientation. We will first look at rotation around the three principle axes (X, Y, Z) and then about an arbitrary axis. Note that for Inverse Rotation: replace q with -q and then R(R-1) = 1

2.4 Z-Axis Rotation



Z-axis rotation is identical to the 2D case:

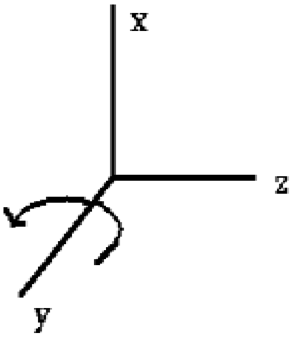
x' = x*cos q - y*sin q

y' = x*sin q + y*cos q

z' = z

$$Rz\,(q) = \begin{pmatrix} \cos q & \sin q & 0 & 0 \\ -\sin q & \cos q & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.4 Y-Axis Rotation



2.4.1 Y-axis rotation looks like Z-axis rotation if replace

X axis with Z axis

Y axis with X axis

Zaxis with Y axis

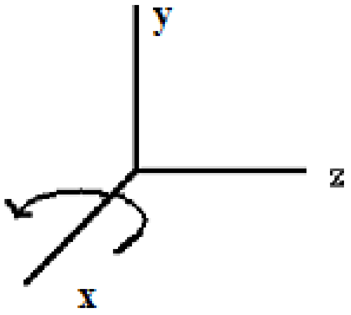
So we do the same replacement in equations :

z' = z*cos q - x*sin q

x' = z*sin q + x*cos q

y' = y

$$Ry(q) = \begin{pmatrix} \cos q & 0 & -\sin q & 0 \\ 0 & 1 & 0 & 0 \\ \sin q & 0 & \cos q & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



2.5 X-Axis Rotation

2.5.1 X-axis rotation looks like Z-axis rotation if replace

X axis with Y axis

Y axis with Z axis

Z axis with X axis

So we do the same replacement in the equations:

$y' = y*\cos q - z*\sin q$

$z' = y*\sin q + z*\cos q$

$x' = x$

$$Rx(q) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos q & \sin q & 0 \\ 0 & -\sin q & \cos q & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

2.6 Rotation about an Arbitrary Axis

This is similar to 2D rotation about an arbitrary point. The general procedure is as follows:

1. Perform transformations which align rotation axis with one of coordinate axis (x, y, z)
2. Perform rotation about the axis
3. Do inverse of (1)

Special case: The rotation axis is parallel to a principle coordinate axis. This is directly analogous to the 2D case of rotation about a point. The steps are:

1. Translate rotation axis to coordinate axis
2. Perform rotation
3. Do inverse translation

In the general case, rotation about an arbitrary axis is more complicated. First we must define the axis of Rotation by 2 points - P1, P2 then do the following:

1. Translate so that rotation axis passes through origin.
2. Rotate so that the rotation axis is aligned with one of the principle coordinate axes.
3. Perform rotation of object about coordinate axis.
4. Perform inverse rotation of 2.
5. Perform inverse translation of 1.

We will arbitrarily choose the Z axis to map the rotation axis onto. The rotation axis is defined by 2 points: P1(x1,y1,z1) and P2(x2,y2,z2.). These 2 points define a vector:

V = (x2 - x1 , y2 - y1 , z2 - z1) = (dx, dy, dz)

Which has a unit vector

U = V |V| where |V| is the length of V = (VxV) = (dx^2 + dy^2 + dz^2)^1/2

Now **U** = (a,b,c) where

a = dx/|V|, b = dy/|V|, c = dz/|V| (these are called the direction cosines of x, y, and z)

(Note: the direction cosine of x = cos A where A = angle of **V** with respect to x axis)

Now we can perform the first translation (of the rotation axis to pass through the origin) by using the matrix **T** (-x1, -y1, -z 1), i.e., move the point P1 to the origin. Next we want to rotate to align **V** with Z axis. We do this in 2 steps:

1. Rotate **V** about X axis to put **V** in XZ plane.
2. Rotate **V** about Y to align with Z.

For rotation about X axis we need to find cos A, sin A where A = angle between projection of **U** (in YZ plane) and Z axis. Note: **U'** is no longer a unit vector, i.e. |U'| =/ 1

Uz = unit vector along z axis = (0,0,1)

now (U)' * (Uz) = |U'| * |Uz| cos A = d * cos A

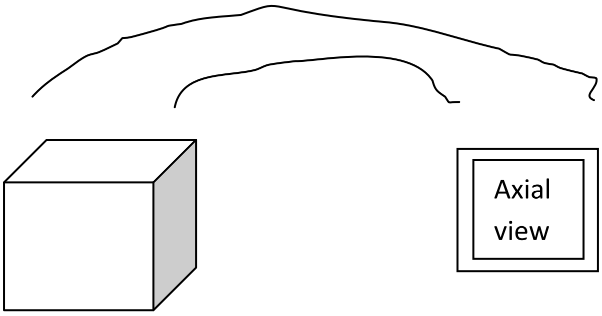
|Uz | = (1)1/2 = 1

(U')(Uz) = 0 * 0 + b * 0 + c * 1 = c

therefore c = d * (cos A) => cos A = c/d

Now the cross product of 2 vectors (V1) X (V2) = W |V| * |V2|sin q where W is perpendicular to plane of V1, V2

so (U') X (Uz) = Ux|U' |* |Uz|sin A = Ux d * sin A



(0 0 0 0)
So Rx(a) = (0 c/d b/a 0)
(0 -b/a c/a 0)
(0 0 0 1)

Rx(a) --> Rotates **U** into XZ plane

Now compute Ry(B) for rotation to z-axis.

After rotation about x-axis the vector is as below:

Uy" = 0 since in XZ plane

Uz" = d = |U'| since just rotated U' into XZ plane

again from dot product:

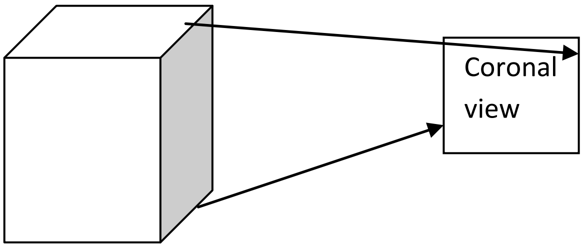
cos B = U" *(Uz) / |U"||Uz| = 0 * a + 0 * 0 + 1 * d = d

Note: |U"|*|Uz| = 1 since both are unit vectors

from the cross product U" x Uz = Uz|U"||Uz|sin B

(from matrix) = Uy x (-a)

therefore sin B = - a

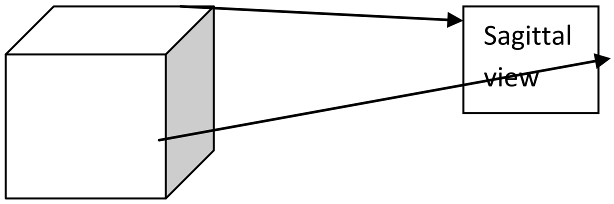


(d 0 a 0)
Ry(B) = (0 1 0 0)
(-a 0 d 0)
(0 0 0 1)

The result of this transformation is that V (= Rotation axis) is coincident with z axis.

Then apply

(cos q sin q 0 0)



$$Rz(q) = \begin{array}{l}(-\sin q\ \cos q\ \ 0\ \ 0)\\ (0\quad 0\quad 1\ \ 0)\\ (0\quad 0\quad 0\ \ 1)\end{array}$$

Then we must apply inverse transformations to get R.A. back to original position. Therefore, the complete composite transformation matrix is as follows:.

$$R(q) = T * Rx(A) * Ry(B) * Rz(q) * RY\text{-}1(B) * Rx\text{-}1(A) * T\text{-}1$$

$$n1 = a(x),\ n2 = b(y),\ n3 = c(z)$$

$$[R] = \left(\begin{array}{lllll} \text{Row1 } n12 + (1 - n12)\cos q & n1n2(1 - \cos q) + n3\sin q & n1n3(1 - \cos q) - n2\sin q & & 0 \\ \text{Row2} n1n2(1 - \cos q) - n3\sin q & n22 + (1 - n22)\cos q & n2n3(1 - \cos q) + n1\sin q & & 0 \\ \text{Row3} n1n3(1 - \cos q) + n2\sin q & n2n3(1 - \cos q) - n1\sin q & n32 + (1 - n32)\cos q & & 0 \\ \text{Row 4 } 0 & 0 & 0 & s & 1 \end{array}\right)$$

III. RESULTS AND ANALYSIS

Generating three views from DICOM images using above methods

- view1 Axial
- view2 Coronal
- view3 Sagittal

still process is going on.

REFERENCE

Books

[1]. Digital Imaging and Communications in Medicine (DICOM) By Oleg S. Pianykh
[2]. Three-Dimensional Imaging for Orthodontics and Maxillofacial Surgery edited by Chung How Kau, Stephen Richmond
[3]. Rotation Transforms for Computer Graphics By John Vince

Websites

[1]. http://www.dcm4che.org/docs/dcm4che-2.0.20-apidocs/org/dcm4che2/data/Tag.html
[2]. http://nipy.org/nibabel/dicom/dicom_orientation.html
[3]. https://www.collabdds.gov.in/index.html
[4]. http://dicomiseasy.blogspot.in/2013/06/getting-oriented-using-image-plane.html
[5]. http://www.dcm4che.org/confluence/display/WEA/Home