Simulating and Reconstructing X-ray Computerized Tomography

Xiang Gao, Qi Li, Suerfu and Yao Zhou

Jan. 15, 2015

1 Project Objective

The main objective of this project is to develop an interface and framework for simulating and reconstructing X-ray computerized tomography. From simple geometric shapes (eg. circles, cubes and spheres) or complicated analytical 2-dimensional and 3-dimensional analytical functions, to real numerical datasets, our program is able to reconstruct the given images with sufficient accuracy and computation efficiency.

2 Background

X-ray is a type of penetrating electromagnetic radiation. X-ray images or X-ray radio-graphs can be obtained by recording the X-ray intensity along the passage of X-ray. In other words, as X-ray passes through the object, the intensity of each point on an X-ray radio-graph is obtained from the line integral of attenuation coefficient of the object.
Although X-ray radio-graphs contains information of reduced dimension, the actual shape and inner structure of an object can be retrieved from them. This is essentially the subject of tomographic image reconstruction. To put it in a more abstract term: we would like to consider the question of whether one can reconstruct 2D (3D) objects from their multiple 1D (2D) images. It was mathematically proven possible by Johann Radon. Later, Godfrey Hounsfield built an X-ray scanner, with which he demonstrated the possibility of reconstructing cross-sectional images of daily objects, for which he was awarded a Nobel prize for physiology.
With the principles and tools that we learned in APC524, we would therefore develop an interface for illustrating the entire process of tomographic image reconstruction. High-performance computers in modern time can handle tomographic image reconstruction much faster than the times of Hounsfield; therefore, it also explains the name X-ray Computerized Tomography (X-ray CT).

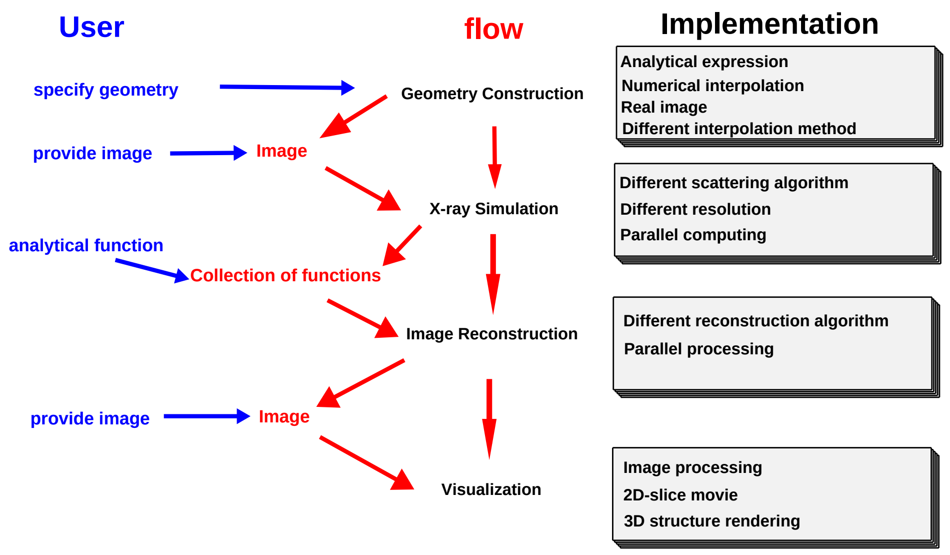


Figure 1: Figure showing how user interacts with the program. The left column shows how user interacts with the program, and the central column shows the flow of the program. The right column is different implementations at each stages.

3 Main Components

The flow of the program is shown in Fig. 1. This section gives a brief description of the main features or principles of each component.

3.1 Geometry Construction:

For the purpose of testing and demonstration, we included simple geometric shapes and analytical functions specified in TestFunctions.cpp. Users can choose from these basic objects to familiarize oneself with the main process of X-ray simulations and reconstruction. Users also have the flexibility to import their own numerical dataset of hdf5-format. Thus, the capability of the reconstruction algorithm can potentially be tested on realistic objects.

3.2 X-ray Simulation:

One X-ray projection of surface A is defined as performing a line integration along the passage of X-ray for a particular angle θ across A. (See Fig.6) One complete round of X-ray projection across A is thus by performing multiple X-ray projections around the central axis of the object. For a 3D object, the

X-ray simulation is done by dividing the object into multiple horizontal slices and obtain one complete round of X-ray projection across each horizontal slice. A useful way to present the information of the resulting X-ray radio-graphs is the 'sinogram'. (See Fig.2) Sinogram is a two-dimensional field of the values of line integration, in which the horizontal axis is the range of one X-ray projection and the vertical axis is the angle at where the projection is taken.

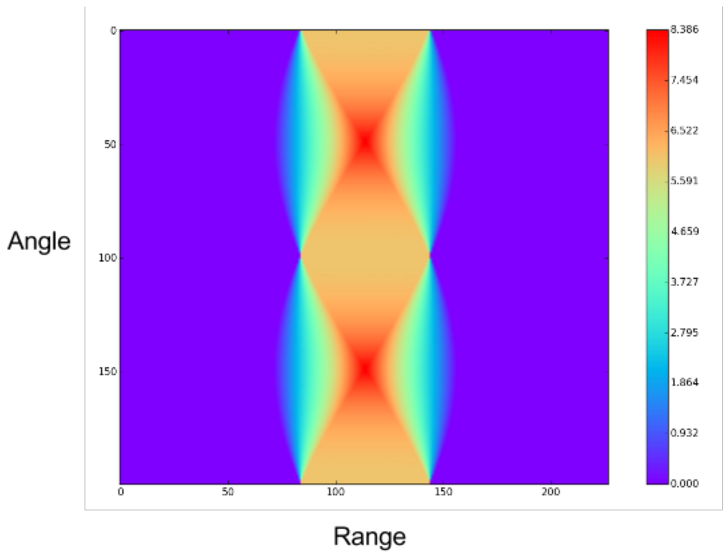


Figure 2: Sinogram of a square shape. Since the square has an axial symmetry of 45 degrees, the sonogram also shows a periodicity of 45 degrees.

This component simulates the actual process of how the intensity of an X-ray radio-graph is generated. The results from performing line integration are then passed to the reconstruction component.

3.3 Tomographic Image Reconstruction:

The program implemented the algorithm called filtered back projection (FBP) to carry out reconstruction. Essentially what FBP does is to take X-ray radio-graphs at various angles θ and height (for 3D objects) and applies a high pass filter to remove blurring that otherwise would have occurred in back projection. Then, the filtered results are back projected. Superposing results of each X-ray scan angle and height (for 3D objects), the original object will be reconstructed. Mathematically, the FBP algorithm can be expressed as :

$$f(x, y) = \int_0^{\pi} d\theta \int_{-\infty}^{\infty} \frac{1}{2\pi} d\omega [|\omega| S_\theta(\omega) \exp(i\omega t)], \tag{1}$$

where f is the value of reconstruction, ω is frequency of the signal $S_\theta(\omega)$, which is the Fourier transform of the X-ray projection at angle θ.
In our program, we use the Hamming window function as the high pass filter, which effectively prevents the blurring but keeping high-frequency noise at a low level.

3.4 Image Visualization:

All the derived classes of Image are equipped with a method named ExportHDF. When the method is called, the data in the class is saved into a designated HDF5 file in directory 'output'. The file includes 1D arrays (/x, /y, and /z, depending on dimension) storing the coordinates of the rectilinear mesh, and array /data storing the value at each node.
A few sample python scripts can also be found in the output directory to visualize the output data (plotCurve.py, plotSurface.py, and movieVolume.py). Note that movie Volume.py requires codec (such as FFMpeg) to save the movie. Example usage: "python plotCurve.py Curve.h5".
When exporting HDF5 files from 2D and 3D Images, a XDMF file is also generated in the output directory to enable reading and visualizing with VisIt (visit.llnl.gov). In VisIt, simply open the .xmf file with format Xdmf and draw contour, pseudocolor, etc.
Besides, all the derived classes of Image also have a constructor from reading in data from a HDF5 file that has the same "flavor" as the output file.

4 Major Classes

This section lists the major classes that are key to the four components described in previous section.

Image Image is the abstract base class with derived classes of one, two and three dimensional shapes, which can either be analytical or numerical. They all implement the method operator () and Print(). X-ray radio graphs are generated by the method called GetProjection. Only Surface and Volume classes need this method since reconstructing a one-dimensional object is trivial. (See Fig.3)

Integrator As explained in the section above, line integration is the central part of obtaining X-ray radio-graphs. It implements the function Integrate, which is passed to GetProjection method to perform the line integral. Different integration methods are available. (See Fig.4)

Interpolator To be able to reconstruct numerical objects successfully, Interpolator is necessary to interpolate the values (i.e. X-ray attenuation coefficient) of the objects when an X-ray scan is performed. In other words, line integration at different angles and height (for 3D objects) requires the X-ray

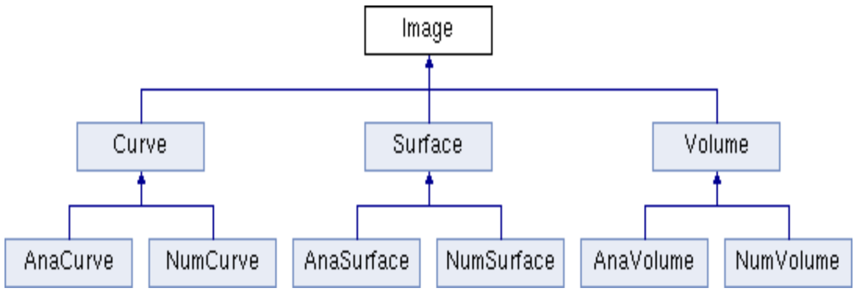


Figure 3: Design structure of Image class and its derived classes.

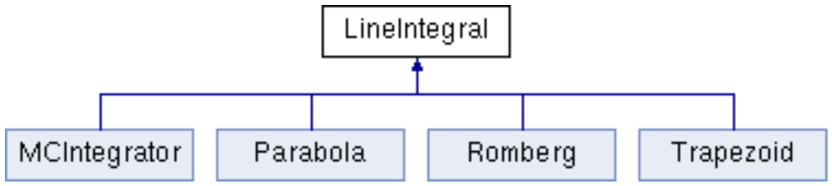


Figure 4: Design structure of integrator class

attenuation coefficient to be interpolated at the coordinates where the numerical objects do not have values due to its discrete nature. (See Fig.5) For most practical purposes in which the numerical data are created on a equally spaced mesh (equally spaced in each direction). Bilinear interpolation is sufficiently accurate. If in case that mesh for numerical objects are not equally spaced, then the NearestNeighborIntpl has to be used.

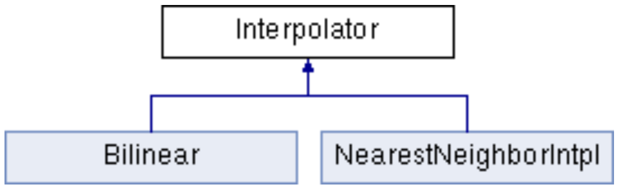
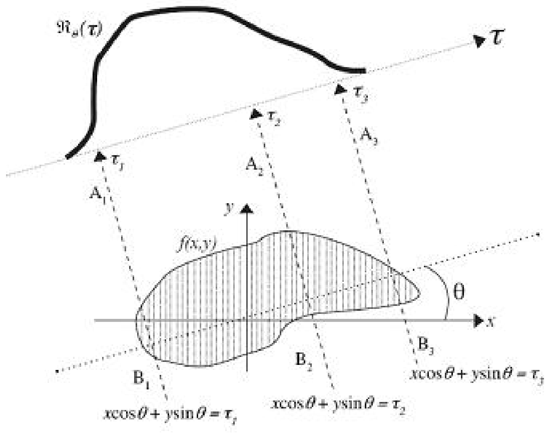


Figure 5: Design structure of Interpolator class.

ImageArray The reconstruction algorithm FBP requires not only the X-ray radio-graphs as the results of line integration but also the corresponding angles and height (for 3D objects). The ImageArray class is an image container that contains the X-ray radio-graph with associated angle and height information as each X-ray scan is generated. The reconstruction algorithm FBP have access to contents of ImageArray during the reconstruction process.


$\Re_\theta(\tau)$
τ
τ_1
τ_2
τ_3
A_1
A_2
A_3
y
$f(x,y)$
θ
x
B_1
B_2
B_3
$x\cos\theta + y\sin\theta = \tau_1$
$x\cos\theta + y\sin\theta = \tau_2$
$x\cos\theta + y\sin\theta = \tau_3$

ht!

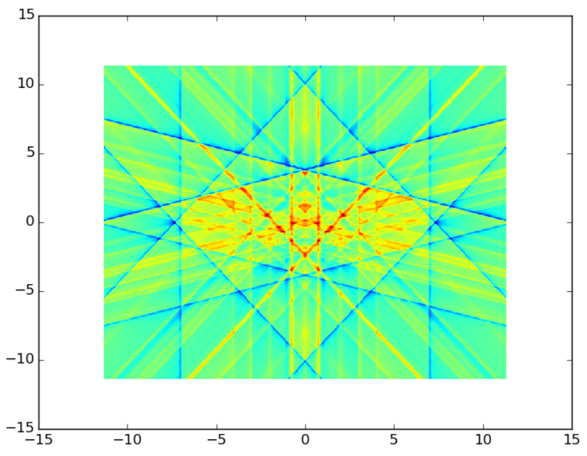
Figure 6: From a 2D scalar field, obtain line integrals along different lines with the same angle[1].

5 Results and discussions

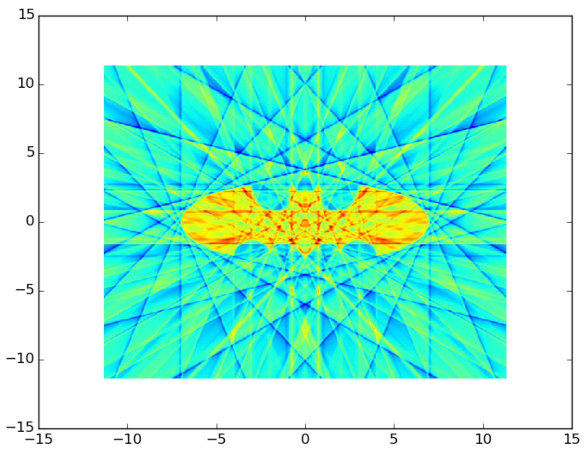
After testing the components separately and having ensured that they work properly, we consider the reconstruction of four main categories to objects. Satisfactory results are obtained with low computational cost. This section presents the results and discussions.
As a general side point regarding the resolution, we found that in general, within a hundred equally spaced angles are sufficient for obtaining the X-ray radiographs. For numerical objects, the spatial resolution is of course limited by the original images for line integration to be carried out. Nevertheless, with a fixed resolution of the original object, it is also quite intuitive that the quality of reconstruction increases with the resolution of one X-ray scan. i.e. the resolution of the line integral directly affects fidelity of reconstruction.

5.1 Reconstruction of 2D Analytical Object

We consider an interesting analytical function (the 'Batman' function) that is constructed from piecewise analytical ones. For all simulations and reconstructions, we consider the domain as a cylindrical shape, such that the radius is able to contain the entire cross-section of the object; the height takes the range of the 3D object.
When the number of equally distributed angles reach ten (See Fig.7b), the rough shape of the 'Batman' becomes visible. When the angle resolution reaches 100, (Fig.7c), the image is very accurately reconstructed with sharp interface.

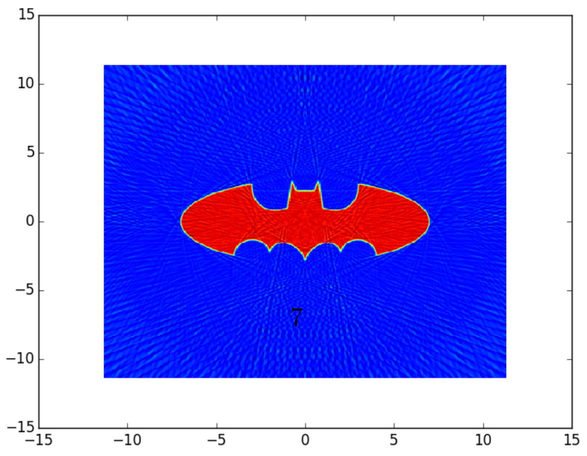


(a) Reconstruction of 'Batman' function from 5 equally spaced angles.



(b) Reconstruction of 'Batman' function from 10 angles.

.



(c) Reconstruction of 'Batman' function from 100 spaced angles.

.

5.2 Reconstruction of 2D Numerical Object

Since the application of X-ray CT is mainly in medical imaging, we considered a numerical sample image of a human spine from MatLab gallery [2] (See Fig.8a). Notice that this image was actually already reconstructed from a real performance of X-ray projection. Nevertheless, we still consider this as a prototype image to test the capability of our program to reproduce this image. The reconstructed one is shown in Fig.8b. The outer structure of the spine as well as magnitude variations of the fine details are well captured. Although some noises are seen which are the 'traces' of back projection, it otherwise validates our program that is able to deal with 'realistic' data sufficiently well.

5.3 Reconstruction of 3D Analytical Object

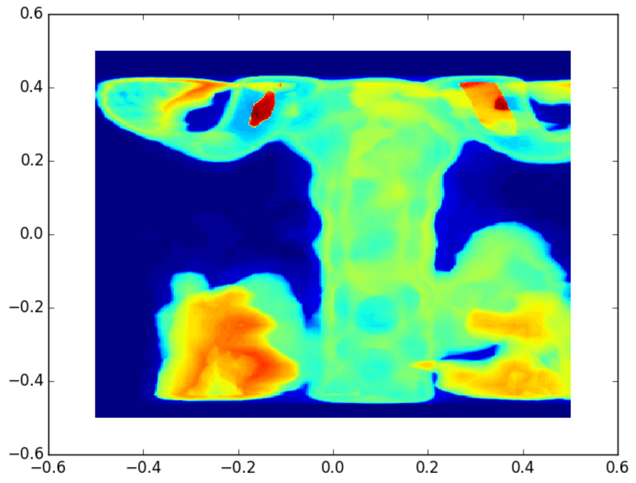
Reconstruction of a three-dimension object is treated as reconstruction from many horizontal slices. Higher resolution in the vertical direction in acquiring the X-ray radio-graphs will of course increase the accuracy of reconstruction. The computational cost of reconstructing a 3D object thus only increases by the multiple of the total number of slices compared to a 2D object.
An interesting analytical 3D object to consider is the heart surface function [3]. The isosurface plot was show in Fig.9a.

5.4 Reconstruction of 3D Numerical Object

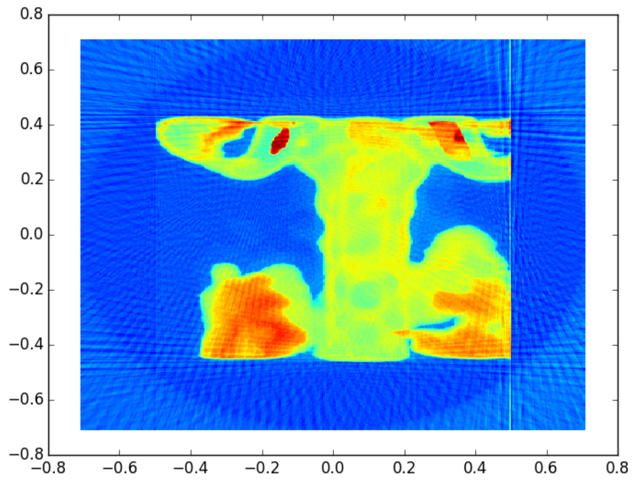
As a test example, we used the MRI three-dimensional dataset provided by MatLab gallery to simulate the reconstruction process. Please see the movies attached.

6 Profiling

We used gprof to profile our software for demoAna2D and demoAnd3D. From the profiling of demoAna2D, we found that as expected Surface::GetProjection was the function called the most number of times. The function takes roughly 96% of entire run time. Within the function call to GetProjection, GetProjectionAtAngle takes up about 96% of time. In GetProjectionAtAngle, Integrate method in line integral takes up 90.5% of time.
FilteredBackProjection takes less than 3 % of total run time. Within FilteredBackProjection the scalar field evaluation method uses about 1.3% of total run time. Compared with this, time spent on convolution/filtering is below 1% and is therefore negligible.
The profiling is similar for demoAna3D with a sphere option. Although the fraction of time call to FilteredBackProjection has slightly increased, the increase is less than 1%, and is negligible. This is expected since the 3D is simply multiple number of 2D slices. So we expect linear complexity.
We conclude that in 2D cases we need to improve GetProjection method in the future versions. Specific measures include better integration method and/or

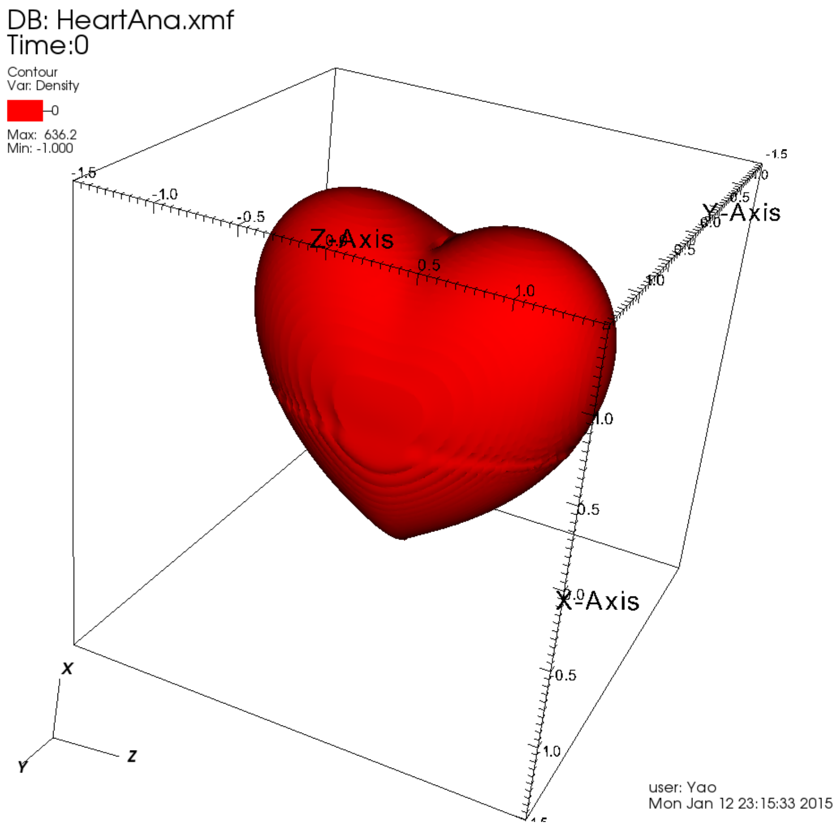


(a) Original image of a human spine.

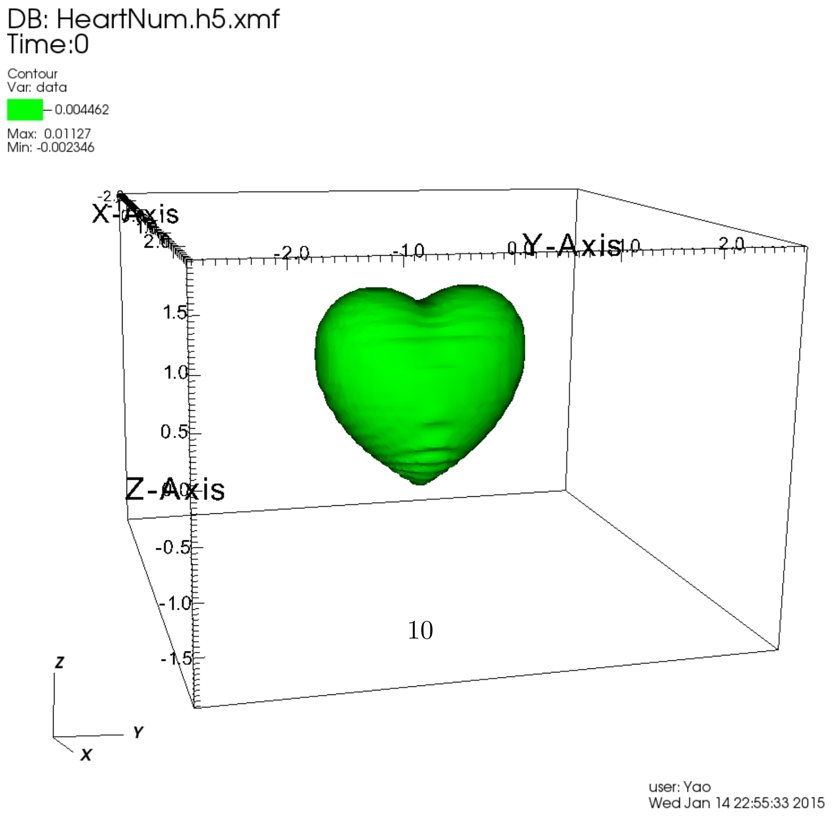


(b) Reconstructed image of a human spine.

.



(a) Original analytical heart function.



(b) Reconstructed analytical heart function parallel computing. Since this portion takes up 96% of total function call, and the process is inherently parallel and independent, parallel computing would almost linearly speed up the software before time spent on GetProjection is comparable to FilteredackProjection().The profile can be found in the git under directory $final/profile_a nanD.txt$.

7 Lessons Learned from APC524

7.1 Designing a good interface is very important

One important principle we learned in the course is programming to an interface. With this principle in mind, the four main components of the program each works independently, and connects to each other through an interface or a protocol.
For example, the geometry component can construct different geometric objects and overload constructors to construct objects in different ways (eg. load a real medical numerical image). The X-ray simulation modules can also call different integration algorithms to generate X-ray radio-graphs.

7.2 Version Control Tool

We used git as the version control tool. Github is the website that we kept track of changes and branches. It a good tool for teamwork and it also encourages one to be more conscious of a good practice of keeping things organized.

References

[1] http://www.mindef.gov.sg/content/imindef/publications/pointer/journals/2008/v34n1/techedge.html

[2] http://www.mathworks.com/discovery/gallery.html

[3] http://www.wolframalpha.com/input/?i=heart+surface