Exercise 1: Cone-Beam CT (CBCT) Simulator

Instructor: Ziv Yaniv (zivy@gwu.edu)

Due date: Feb. 16 2012

In this exercise you will implement a CBCT simulator. The program can be written either in C++ or using MATLAB. You receive three pieces of information which comprise our data set:

1. Projection images in DICOM format and in the Insight Segmentation and Registration Toolkit (ITK) meta image format[1].
2. The reconstructed volume in DICOM format and in the Insight Segmentation and Registration Toolkit (ITK) meta image format.
3. A text file containing the projection matrices associated with each of the projection images.

You will implement the following two programs:

1. CBCTGenerateProjections - receives a volume and generates the projection images from a rotation.
2. CBCTReconstruct - receives a set of projection images and reconstructs a volume using the SART algorithm discussed in class.

You will also generate an additional volume and corresponding projection images, all in meta image format. The volume should span the same spatial extent as that provided to you. The content of the volume is up to you. The simplest acceptable volume contains a single sphere centered at an arbitrary location.

Submission:

1. Report in pdf format describing your code, identifying bottlenecks in the implementation and any modifications you implemented to improve the computational complexity. Report should also include pictures of results generated by your two programs and pictures of the new volume and projection data you

[1] http://www.itk.org/Wiki/MetaIO/Documentation

created. Also, provide the normalized correlation coefficient between your reconstruction and the original volume, both for the data I provided and the data you provide.

2. Your source code and instructions on how to compile (C++) and how to run (C++ and MATLAB implementations).
3. Your volume and projection images.

Words of caution:

- Developing the programs using data at full resolution is extremely time consuming due to the computational complexity of the algorithms - sub-sample and then sub-sample some more.
- Use the ITK-SNAP (http://www.itksnap.org) program as a viewer for visual debugging.