Pre-defined project #1: ray-casting

Implement a ray-casting volume renderer that uses compositing for its ray function. The volume renderer should work on rectilinear grids, should be able to cast rays using perspective projection (i.e., like the slides in class) from arbitrary camera positions. I will provide specifications (a data set, transfer functions, camera positions, and image size) on the last week of instruction (i.e., Dec. 2nd-6th). In the interim, you can use the scalar “hardyglobal” from proj7.vtk. The data set provided during the last week of instruction will also be in the “.vtk” format, so it should be integrate into your project. Your camera should be in the same format described in lecture.

You should work with the data structures for a camera and a transfer function posted to the web as part of the project, as that is how I will specify the final images to make.

Your deliverables are three-fold:
(1) your code
(2) the still images you produce following the specifications I provide on the last week of instruction
(3) a table showing the performance impact of early ray termination for these images

The still images you produce will be shown during the class final.

Pre-defined project #2: make a movie

Make a movie using VTK or using VisIt. I will provide example data by Tuesday, Nov 19th. You already know how to use VTK. I will lecture on VisIt on Wednesday. Further, there is an online tutorial on VisIt: http://visitusers.org/index.php?title=Short_Tutorial

The movie will consist of many still frames that you will encode into a video format (i.e., MPEG). You movie should animate things in the data set ... isovalue, colors, camera position, etc. It will be graded on how polished the movie appears, and how well you convey the nature of the data.

The movies will be shown during the class final.