CPSC 641 – Computer Graphics
Assignment 4 due 12/5/17 (class time)

The goal of this assignment is to give you practice implementing an alternative rendering approach. You have a choice among eight options, and should pick one of these (you can pick any of these options). Note that if you have previously implemented one of these for a class, then that particular one should not be chosen as an option for this assignment.

Here are the options for your implementation:

1. **Volume Rendering:**
   There are three options for volume rendering. In each case, you should build your own rendering system that will clearly display the volume, and allow views from multiple directions (you can either change the eyepoint, or rotate the volume itself).
   a. **Isosurface Rendering:** You should create an isosurface from the volumetric data, and render that isosurface. You should implement your own version of marching cubes (or some other isosurfacing method), and should try to set normal so that the final rendering appears smooth.
   b. **Splatting:** You should create a rendering using a splatting approach. Your splatting algorithm should include defining a transfer function, warping of individual splats, lighting of the splats, and blending as appropriate.
   c. **Volumetric Scattering:** You should create a ray-traced or path-traced rendering through a varying volume, in which you compute light scattering along the ray path. You need to account for single-scattering, but not necessarily multiple scattering. You can use any reasonable (i.e. realistic in that it is in widespread use) scattering function for determining scattering.
2. **Nonphotorealistic Rendering:**
   a. **Gooch Shading/Toon Shading/etc.:** You should implement a real-time rendering scheme that implements Gooch shading, Toon shading, or some similar method. The method used should involve both changing the color rendering within some model, and performing edge highlighting. You can choose from any method for highlighting/darkening edges that you wish.
   b. **Painterly rendering:** You should implement a method that modifies the output image by creating a brush-stroke-based effect. You can use a standard rendering approach to create an initial image, but must then process the image to one that gives a painted effect. The effect must include some sort of stroke-based processing (not just a recolorization/blurring/etc.) to create an image that has some appearance of having been painted.
   c. **Pen-and-ink or similar rendering:** You should implement a method for generating images that look similar to pen-and-ink illustrations. The

expectation here is that you would include methods that would use varying mipmap textures to shade objects at some level of zoom, along with some method for drawing key edges. You do NOT need to perform all aspects of a pen-and-ink approach (e.g. giving different tone selection, for instance, or all of the details normally included in stroke specification). I am expecting something that highlights edges with strokes and that uses some sort of mipmap-like levels to draw different hatching patterns on surfaces, though.

3. **Image-Based Rendering:**
    a. **Lightfield rendering:** You should create a lightfield representation of a scene – you can precompute this from various ray casts from a base configuration if desired. You should then allow someone to navigate through the lightfield to find different views of the scene. For maximum credit, you should try to use the lightfield information to allow rendering differing depth of field views.
    b. **Multi-image warping:** You should implement a method that uses multiple images to create views that are warps between them. You should use at least two images, and demonstrate a continuous shift of viewpoints in which those views are used to define the eventually viewed geometry.

Notes:

1. If you have a different variation that you would like to implement that still falls into one of those broad categories (Volume rendering, NPR, or IBR), then please ask; it might be possible to have a different implementation.
2. It should be relatively easy to find references to each of these techniques online. Some of them are well-established ideas, while others are best described by a single paper (and usually subsequent discussion of that paper). If you have a particular method you are interested in and need a pointer to start on it, please ask!
3. You will need to submit, on the final day of class, a document that includes the following:
    a. A statement about which method you chose to implement.
    b. A list of resources you used to obtain input (e.g. the source for the volume or image data you used).
    c. Any other notes regarding the development or use of your software – external libraries used, resources needed to run, etc.
4. You should plan to demonstrate your assignment in the 3 days following the due date. A schedule for demos will be announced.