Submission instructions: You should submit a hard copy of your work in class, and upload your code to Stellar. If submitting late, please upload a PDF to Stellar together with your code. **For this assignment, please submit *simpleWorldY.m* ONLY, and any new images you might have taken**.

Lecture 1 notes: `http://6.869.csail.mit.edu/fa13/lectures/chapter_01_simplesystem.pdf`

Code: `http://6.869.csail.mit.edu/fa13/psets/pset1.zip`

Stellar: `https://stellar.mit.edu/S/course/6/fa13/6.869/`

Problem 1: *A simple image formation model (optional, extra credit: 1 late day)*

The goal of this first exercise is to take images with different settings of a camera to create pictures with perspective projection and with orthographic projection. Both pictures should cover the same part of the scene. You can take pictures of real places (e.g., the street, a living room, etc) or you can also create your own simple world (e.g., you can print *simpleWorld.pdf*, included in the code package, and create your own scenes. I recommend printing on matte paper).

To create pictures with orthographic projection you can do two things: 1) use the zoom of the camera, or 2) crop the central part of a picture. You will have to play with the distance between the camera and the scene, and with the zoom (or amount of cropping) so that both images look as similar as possible only differing in the type of projection (similar to Figure 1.4, in the Lecture 1 notes). Submit the two pictures.

Problem 2: *Orthographic projection*

Derive the projection equations (Eq. 1.2 and 1.3 in Lecture 1 notes) that relate the coordinates of a particular point in the 3D world and the image coordinates of the projection of that point in the camera plane (in the code, we assume $\alpha = 1$, but do not use this assumption in your solution).

Problem 3: *Constraints*

In Lecture 1 notes, we have written all the derivative constraints for $Y(x, y)$. Write the constraints for $Z(x, y)$. We will use the constraints on Y in the code, as specified in Problem 4.

Problem 4: *Approximation of derivatives*

Fill in the missing kernels (lines 166 and 180) in the script: *simpleworldY.m*. Show the code you filled in these two lines in your writeup (together with uploading it to Stellar).

Problem 5: *Run the code*

Select some of the images included with the code and show some new view points for them.

Optional: You can also try this with new images of your own 'simple world'.

Problem 6: *Violating simple world assumptions*

Find an example from the images provided with the problem set or from one of your own images where the recovery of 3D information fails. Explain why it fails.

Problem 7: *Research problem (optional) - The real world*

The goal of this problem is to test the 3D reconstruction code with real images. A number of the assumptions we have made will not work when the input is real images of more complex scenes. For instance, the simple strategy of differentiating between foreground and background segmentation will not work with other scenes.

Try taking pictures of real world scenes (not the cubes) and propose modifications to the scheme proposed in this lecture so that you can get better 3D reconstructions. The goal is not to build a general system, but to be able to handle a few more situations.