**COMP-558: Assignment 2**
**Available: Tuesday, February 19th, 2013**
**Due Date: Monday, March 11th, 2013**

**Notes:** You should use **MATLAB** to carry out this assignment. Be creative (but careful) in presenting your results and in performing the numerical comparisons. I expect everyone to submit original work. This means that *you must write your own code and any results you hand in should be based on your own implementations.* Be sure to submit your Matlab code along with your answers to the questions below. Since this assignment requires a fair amount of experimental work one of the things we will ask you to do is demo your results to the TAs during office hours, the week of the due date. This will help us assess the quality of your work.

**Question I:**[40 marks]

Consider the eight point algorithm we have discussed in class (section 7.3.5). On Emmanuel's COMP 558 page which you can get to from `http://www.cim.mcgill.ca/~siddiqi/558b.html`, we will be posting two pairs of images, which compose two stereo pairs of two different 3D scenes. Implement Algorithm **EIGHT-POINT** (page 156) to estimate the fundamental matrix $F$ for each of the two stereo pairs. To do so, in each stereo pair, you will have to manually identify $n$ corresponding pixel pairs. Be sure to choose correspondences that seem visually unique. Pay attention to the procedure outlined at the end of Section 7.3.5 to avoid numerical instabilities. Explain your working, and examine how "stable" the entries of $F$ are, as the pixel pairs are changed, and as the value of $n$ changes. Explore this experimentally, reporting on your results.

Implement algorithm **EPIPOLES-LOCATION** (section 7.3.6, page 157) and compute the locations of the epipoles in each image, for both stereo pairs. Your output for this question should be the two fundamental matrices, the location of the epipole for each image, as well as a discussion of your results. Be sure to hand in your source code for the two algorithms. *Hint*: Computing SVD is trivial in Matlab.

**Question II:**[30 marks]

Now assume that the matrices of "intrinsic" parameters of the two images in each pair are:

$$M_r = M_l = \begin{bmatrix} -f/s_x & 0 & o_x \\ 0 & -f/s_y & o_y \\ 0 & 0 & 1 \end{bmatrix},$$

where the pixels sizes are $s_x = s_y = 31.2 \times 10^{-6}$m, and you can assume that the image center (the principal point) $(o_x, o_y)$ is actually located in the center of the image. For stereo pair 1, the focal length $f = 31$mm, and for stereo pair 2, the focal length $f = 32$mm. Implement Algorithm **TRIANG** (section 7.4.1, page 162) and use it to recover the depth of several corresponding pixel pairs chosen manually, in each stereo pair. For each stereo pair, choose your corresponding pixel pairs so that they belong to at least 5 different objects, so that you can recover at least 5 different depth levels.

Indicate the resulting depth on the input images, for each pixel pair that you used. Do that in a concise and meaningful way. For example, you can try to use color coding for depth, or some other visualization scheme in Matlab. Hand in your source code, as well as a discussion of your work and your results.

**Question III:**[30 marks]
You will now use the code developed above to reconstruct a 3D scene from a stereo pair of your choice. Create a scene with an interesting object (or objects) with surface texture to allow for dense correspondences to be computed. Obtain a stereo pair by taking two photographs of the scene with a digital camera, where the second photograph is taken from a slightly different vantage point (a small translation and a rotation) than the first. Now carry out the following steps:

1. Using your implementation in Question I, estimate the fundamental matrix for your stereo pair. As before, you can do this by using a certain number of fixed correspondences, determined by hand.

2. Write a simple procedure to compute matching locations between corresponding points in the left-right image pairs. You can do this by using the process we discussed in class, i.e., for each pixel in the left image obtain its match by selecting the pixel in the right image (searching over a neighborhood of "nearby" locations) that minimizes the sum of squared differences in intensity within a window. For this choose a window size and a search neighborhood that seems appropriate for your scene. Visualize your correspondence results to show that they are meaningful. Note: for clarity do not show correspondences at every pixel, but rather, for a subset of pixels obtained by subsampling the original left-right image pair.

3. Making reasonable assumptions about the focal length, the pixel spacing and the camera center (you can treat these as inputs to the **TRIANG** algorithm implemented in Question 2) and the dense set of correspondences found above, try to reconstruct the underlying 3D scene. You can use any visualization tools available in Matlab to show the reconstructed (3D) scene points. Discuss your results.