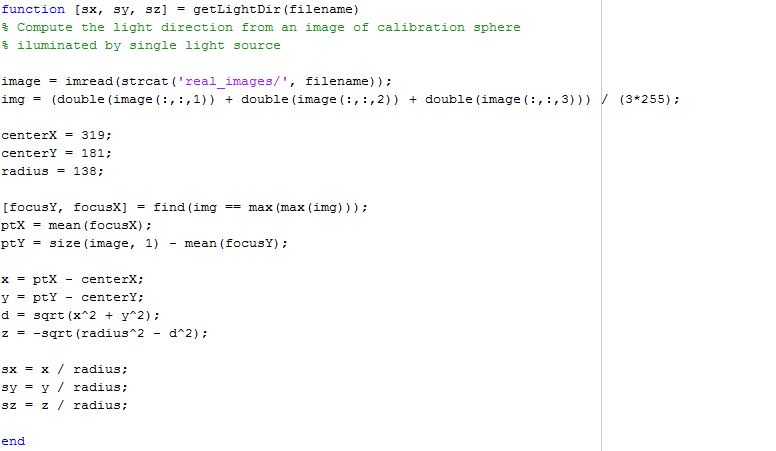
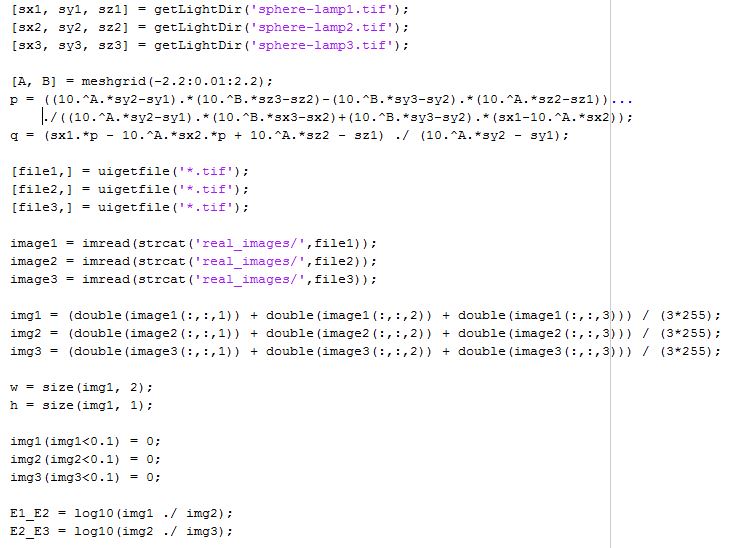
CMPT412 A3 Documentation

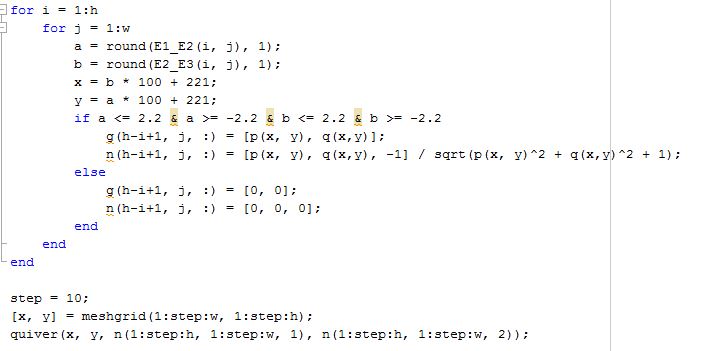
**Code:**

getLightDir.m



PhotometricStereo.m





**Method Description:**

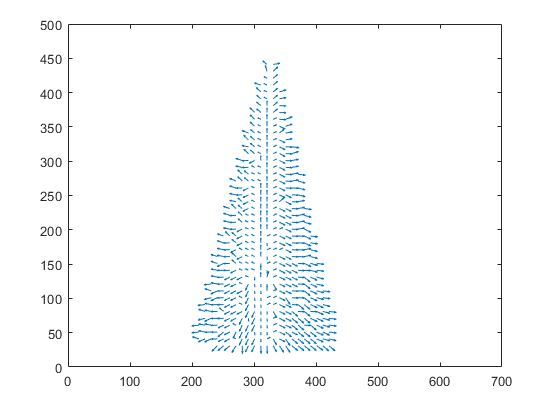
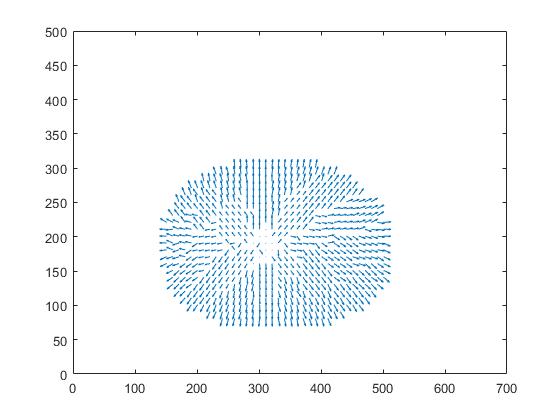
All methods I have used in the assignment are simple algorithms from either the lectures or notes given.

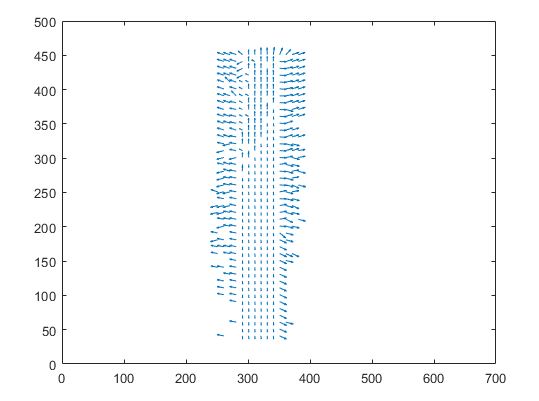
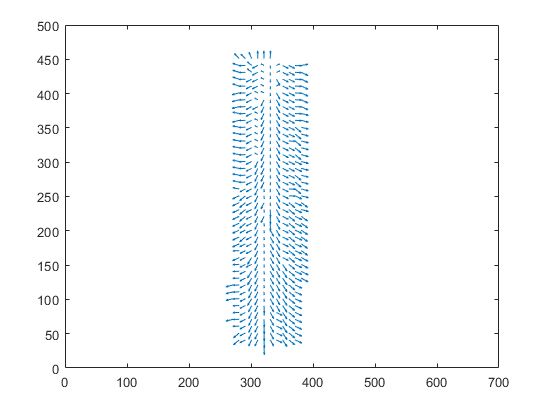
As mentioned by the professor, the very first step is to calculate the directions of the three light source, from the images of a calibration sphere illuminated by particular light at different locations, one light for one image. The key point is to find the brightest spot on the sphere surface, and then apply the Pythagoras theorem to find the Z value of the light vector when viewing the sphere in a 3D scene, with the center location and the radius obtained in advance (which is easy to guess from the images provided). Since we assume all objects are lying in the left-hand coordinate system, all light direction vectors going through the sphere center and the brightest point should have a negative Z values.

To find the surface normal of any object at each pixel, I strictly followed step 4 and 5 in the assignment instruction:

1. First derive the equation for gradient components *p* and *q*, with respect to the intensity ratio and , from the known equation for intensity as a function of gradient. The benefit of using the ratio of two intensities instead of an intensity itself is that we can eliminate the surface reflectance factor as the third unknown in the intensity-gradient function. Also we add a logarithm in which a larger range of ratio can be represented by an equally spaced values.
2. Then we have two lookup tables for *p* and *q* respectively if we apply the two derived equations above into a meshgrid whose rows and columns indexing every pair of the two ratios, with reasonable range (eg. from 1/256 and 256/1), since zero intensity is not considered.
3. Then read in the three images of a particular unknown object, compute the intensity ratio for each pixel. Then for each pixel, we can search in the lookup table for *p* and *q* using the two intensity ratio as index x and y. As well, the surface normal for each pixel can be calculated using the corresponding gradient.

**Example Output (all for real\_images):**

****

****

**Comment:**

As can be seen, the calculated normal shown in the result quiver plots are not very accurate to the true normal of the object, so there might be some limitations of my method.

Firstly, in case that the light source directions are not given and we must calculate them from the images, there might be errors between the true directions and the computed ones, considering many factors that will affect the intensity displayed on pixels, such as the condition of the light source and whether the sphere surface is ideal or not so that the reflection will be accurate for every pixel.

Secondly, the images of unknown object are typically not perfect. There might be ambient light or other radiance caused by the scene environment that introduce complexity and deviation in calculation, but all my method and algorithm are based on the assumption that the images are taken in a perfect condition. Thus errors and incompleteness are unavoidable due to my too simple method.