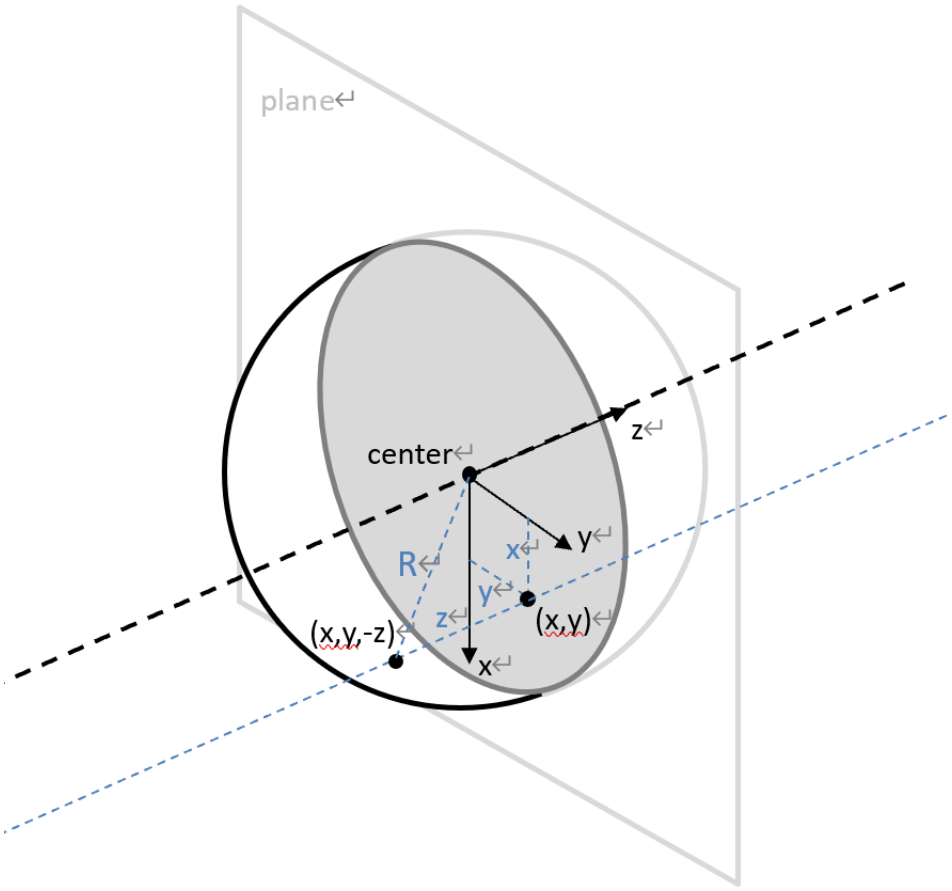
# Computer Vision Homework 4 README


## Jiangtian (Joey) Hou, jh4170


Challenge 1a worked fine. I used help from official MATLAB website to use the function regionprops(). I used stats = regionprops('table',bw,'Centroid','MajorAxisLength','MinorAxisLength') for the center and the diameter (and radius by dividing the diameter by 2).

Challenge 1b worked fine. Below is an explanatory figure of how I found the formula of the normal vector formula of a pixel on the graph:




If the coordinate of the pixel is $(x_p, y_p)$ and the center of the circle is $(x_0, y_0)$, we have

$$x = x_p - x_0$$

$$y = y_p - y_p$$

and since

$$R^2 = x^2 + y^2 + z^2$$

[REQUIRED] The formula of the normal vector is

$$\mathbf{n} = \begin{bmatrix} x & y & -z \end{bmatrix}$$

The negative sign of z-coordinate is because of the positive direction of z-axis is facing inside the plane. The positive direction of z-axis is decided by the right-hand rule.

[REQUIRED] If the normal vector needs to be normalized, the formula is:

$$\hat{\mathbf{n}} = \frac{1}{||\mathbf{n}||}\begin{bmatrix} x & y & -z \end{bmatrix} = \frac{1}{R}\begin{bmatrix} x & y & -z \end{bmatrix}$$

[REQUIRED] If the normal vector needs to carry the magnitude that is equal to the brightness at that pixel, the formula is:

$$\mathbf{n}_{\text{brightness}} = \text{Brightness} \cdot \hat{\mathbf{n}} = \frac{\text{Brightness}}{R}\begin{bmatrix} x & y & -z \end{bmatrix}$$

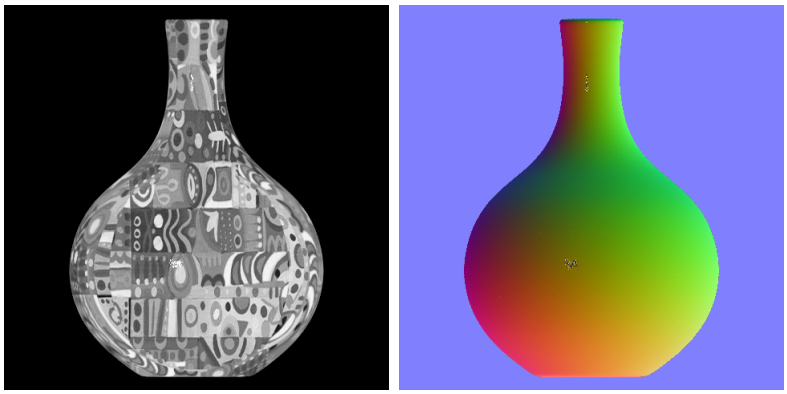
[REQUIRED] For the reason of deciding the direction of normal vectors: Since the intensity is proportional to $\mathbf{n} \cdot \mathbf{s}$ (for lambertian surfaces), when the angle between $\mathbf{n}$ and $\mathbf{s}$ is the smallest (that is, 0), the dot product is the greatest and the intensity will also be maximized. Therefore, the brightest point correspond to the greatest intensity, and it is on the line connecting the light source and the center of the sphere.

Challenge 1c worked fine. I added all five images together, pixel by pixel, and then mark those non-zero sum pixels to be 1 (and the rest of the pixels are all 0 and thus are a part of the background).

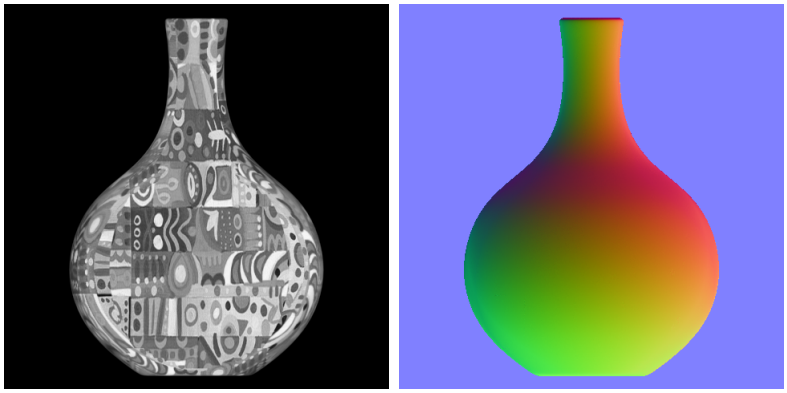
Challenge 1d worked fine. I wrote two versions of using light sources: (1) 3 brightest light sources; and (2) all 5 light sources. For (1), I used maxk() to get the 3 brightest pixels at the same position and then determine matrices S and I based on this. For (2), I just calculated normal vectors by constructing a 5x3 matrix S from all 5 light sources. Since (2) worked better than (1), I commented out codes for (1).

The result of (1) and (2) are displayed here:

(1) There are noises (and the coordinates are incorrect but it is because of other reasons).



(2) They look clear and complete (the coordinates are correct).



All files, original or generated, are included in the zip file.