

RBE549 - Homework 2

Keith Chester

Due date: September 8, 2022

Problem 1

In this problem, we attempt to prove with similar triangles that:

$$\frac{1}{z_0} + \frac{1}{z_c} = \frac{1}{f} \quad (1)$$

First, we look at the original diagram provided. I have added markers for the lengths (z_0 and z_c) assigned to their lengths, and marked the two lengths of f to its distance from the center of the lens.

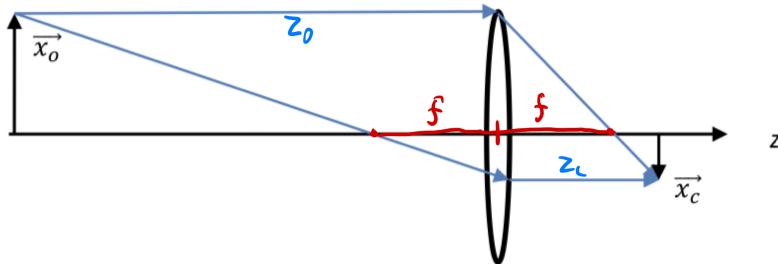


Figure 1: Problem 1

We will note that the triangle provided by z_0 and be represented on the other side of the lens by flipping it along the lens as an axis. From there we can begin to mark angles.

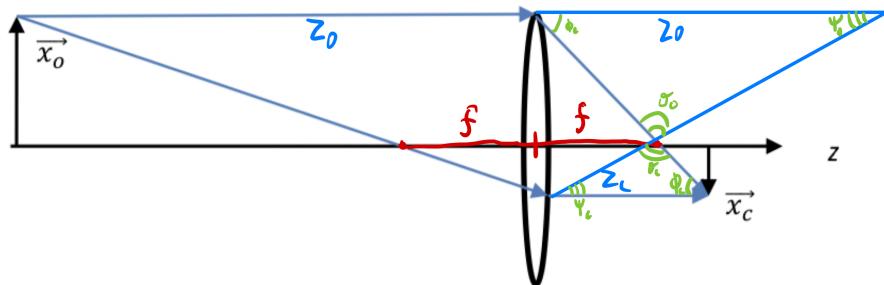


Figure 2: Problem 1 Expanded

Because the z access, z_c , and z_0 are parallel lines, we can thus know that opposite angles of lines crossing them are equivalent. Thus we see that ϕ_c and ϕ_0 are equivalent and marked as such. The same goes for ψ_c and ψ_0 , as marked. Since the lines passing through the focal point f is from parallel lines and continuous, we can also determine that θ_0 and θ_c are also equivalent.

With this, we know that these are similar triangles, and can begin to form ratios to represent the relationships. Thus we have:

$$\frac{z_0}{z_c} = \frac{f}{z_c - f} \quad (2)$$

From here we can perform algebra to isolate the variables into a manner that is closer to what we sought to originally prove:

$$z_0 z_c - z_0 f = z_c f \quad (3)$$

$$z_0 z_c = z_c f + z_0 f \quad (4)$$

$$z_0 z_c = f(z_c + z_0) \quad (5)$$

$$\frac{z_0 z_c}{f} = z_c + z_0 \quad (6)$$

$$\frac{1}{f} = \frac{z_c + z_0}{z_0 z_c} \quad (7)$$

$$\frac{1}{f} = \frac{z_c}{z_0 z_c} + \frac{z_0}{z_0 z_c} \quad (8)$$

$$\frac{1}{f} = \frac{1}{z_0} + \frac{1}{z_c} \quad (9)$$

Problem 2

A

A typical human eyeball is 2.4cm in diameter and contains roughly 150,000,000 receptors. Assuming a uniform distribution of receptors across the hemisphere, how many receptors are there per mm^2 ?

$$\text{Area of hemisphere} = \frac{1}{2} 4\pi r^2 = 2\pi r^2 \quad (10)$$

$$= 2\pi \left(\frac{0.24}{2}\right)^2 = 2\pi 0.12^2 = 904.78mm \quad (11)$$

$$\frac{150,000,000}{904.78} \approx 165,786 \quad (12)$$

So with these assumptions there are approximatley 165,786 receptors per mm^2 .

B

In this problem, we put forth that mars mars has a diameter of 8,000km and an average distance from Earth of 225,000,000km. Using a value of f equal to the eye's diameter, on how many receptors does the image of Mars fall?

We can use similar triangles, as we did in problem one, to create a ratio which we can use to determine the size of the image of Mars on our eye, and from this determine the number of receptors that see Mars. The ratio we ultimately aim to achieve first is:

$$\frac{z_0}{z_c} = \frac{h_0}{h_1} \quad (13)$$

...but what is z_c ?

As we saw earlier, we can use:

$$\frac{1}{f} = \frac{1}{z_0} + \frac{1}{z_c} \quad (14)$$

$$\frac{1}{2.4cm} = \frac{1}{225,000,000km} + \frac{1}{z_c} \quad (15)$$

$$z_c = 2.4cm \quad (16)$$

...Which is a barely noticeable value smaller than our given focal length. With this, we can now set up our ration of image sizing:

$$\frac{8,000km}{h_i} = \frac{225,000,000km}{2.4cm} \quad (17)$$

...and the height/diameter of our mars image is thus:

$$h_i = 8.53 * 10^{-7}m \quad (18)$$

Multiplied by our receptors per mm caluclated earlier:

$$8.53 * 10^{-7} \times 165,786 \approx 0.141 \text{ receptors} \quad (19)$$

...this results in less than one receptor receiving the light of mars, which means we can't see it normally with our eyes alone.

Problem 3

An image has object and background pixels whose brightness values are distributed in accodance to a Gaussian (normal) distribution with the same mean μ but different variances σ_o and σ_b with $\sigma_o < \sigma_b$.

$$P_o(x - \mu) > P_b(x - \mu) \quad (20)$$

It is desired to segment the image into object and background. To this end we wish to show that the decision rule that maximizes the probability of a correct decision is to label each pixel with brightness x as *object* or *background*:

$$\frac{1}{\sqrt{2\pi\sigma_o}} e^{\frac{-(x-\mu)^2}{2\sigma_o^2}} > \frac{1}{\sqrt{2\pi\sigma_b}} e^{\frac{-(x-\mu)^2}{2\sigma_b^2}} \quad (21)$$

First we will try to isolate our terms by taking the natural log. Since $\ln(ab) = \ln(a) + \ln(b)$ and $\ln(\frac{a}{b}) = \ln(a) - \ln(b)$...:

$$\ln(1) - \ln(\sqrt{2\pi\sigma_o}) + \frac{-(x-\mu)^2}{2\sigma_o^2} > \ln(1) - \ln(\sqrt{2\pi\sigma_b}) + \frac{-(x-\mu)^2}{2\sigma_b^2} \quad (22)$$

$$\ln(\sigma_b) - \ln(\sigma_o) > \frac{-(x-\mu)^2}{2\sigma_o^2} - \frac{-(x-\mu)^2}{2\sigma_b^2} \quad (23)$$

$$\ln(\sigma_b) - \ln(\sigma_o) > \frac{((x-\mu)^2 2\sigma_b^2 - ((x-\mu)^2 2\sigma_o^2)}{2\sigma_o^2 2\sigma_b^2} \quad (24)$$

$$(\ln(\sigma_b) - \ln(\sigma_o))(2\sigma_o^2 2\sigma_b^2) > ((x-\mu)^2 2\sigma_b^2) - ((x-\mu)^2 2\sigma_o^2) \quad (25)$$

$$\frac{(\ln(\sigma_b) - \ln(\sigma_o))(2\sigma_o^2 2\sigma_b^2)}{\sigma_b^2 - \sigma_o^2} > (x-\mu)^2 \quad (26)$$

...and finally we can finalize this result by taking the square root of above, resulting in:

$$\sigma_b \sigma_o \sqrt{\frac{\ln(\sigma_b) - \ln(\sigma_o)}{\sigma_b^2 \sigma_o^2}} > |x - \mu| \quad (27)$$

Problem 4

In this problem, we look at the inverse of a transformation matrix, H , where:

$$H = \begin{bmatrix} \mathbf{R} & \vdots & T \\ \dots & & \dots \\ 0 & \vdots & 1 \end{bmatrix} \quad (28)$$

The true inverse of a transformation matrix is:

$$H^{-1} = \begin{bmatrix} \mathbf{R}^T & \vdots & -\mathbf{R}^T T \\ \dots & & \dots \\ 0 & \vdots & 1 \end{bmatrix} \quad (29)$$

This additional step is needed as the translation from the original point is influenced by the orientation of the previous transform; therefore we must perform a translation that also considers reversing the affected rotation as well.

Problem 5

In this problem, we attempt to create a binary threshold of a selfie image in order to demonstrate thresholding. The image in question is below.



Figure 3: Pop up dinner selfie

A

In this section, we use OpenCV to open the image and convert it to grayscale. Once we have the image in grayscale, we ask OpenCV to create a histogram of bins, where each possible pixel value (from 0 to 255, or one byte) is a bin. From here, we check how many of the pixels exist in each bin. Once we have hit 80% of the image's pixels being set to white, we are done.

The code for this can be viewed in *problem_5a.py*. Below is the image generated from the code.

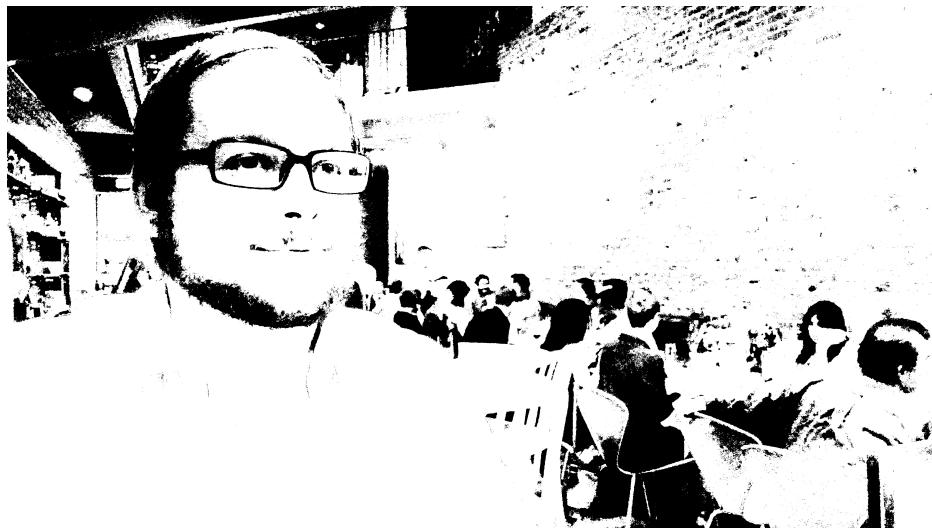


Figure 4: Binary Threshold

B

For this section, we wish to experiment with other thresholding types. The code for this can be viewed in *problem_5b.py*. Instead of building the threshold out manually, we use OpenCV's built in thresholding functions. We demonstrate a straight binary threshold (similar to what we did before, but with a cutoff value of 127, or half max intensity) and then a mean and gaussian adaptive threshold. These methods are utilized to demonstrate common methods that one would have already prepared for them in the toolbox of OpenCV. Note that I performed a gaussian blur on the image prior as well, in hopes to reduce noise on these thresholding functions.



Figure 5: Binary Threshold

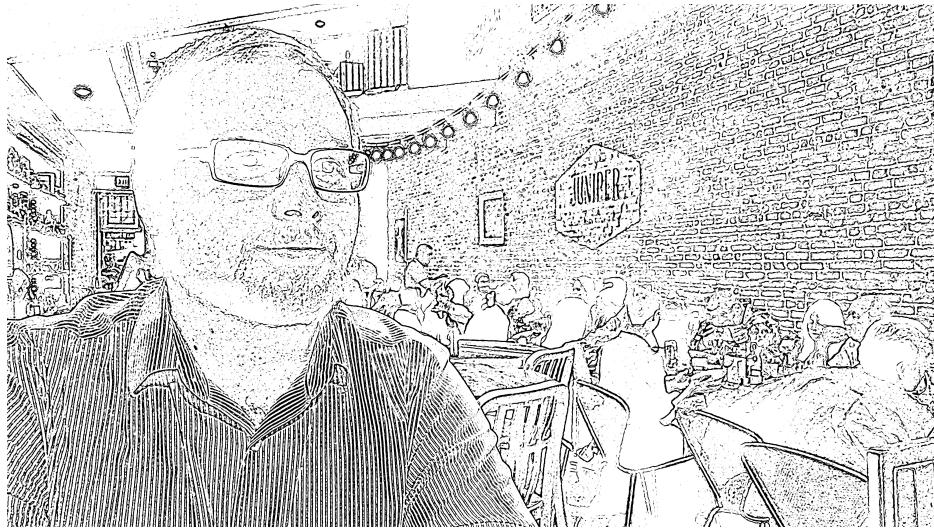


Figure 6: Adaptive Threshold - Mean Blur

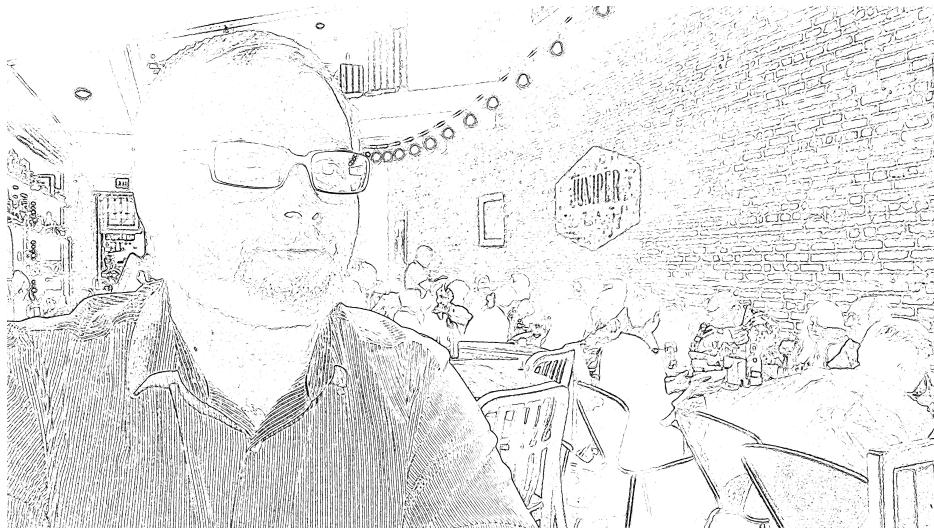


Figure 7: Binary Threshold - Gaussian Blur