Focusing & Identifying Images with Ray Tracing

Sherlock Zhang
Introduction to Electrical Engineering
Washington University in St. Louis
St. Louis, United States
jiaxi.zhang@wustl.edu

Brant Yang
Introduction to Electrical Engineering
Washington University in St. Louis
St. Louis, United States
boran.y@wustl.edu

***Abstract*— In this project, we developed a model that simulates the rays of light traveling in free space, transformed by a thin lens, then creates a clear, sharp image at a certain distance on the other side of the lens. The simulation models how optical imaging systems in our daily lives captures light rays and turns them into images that we see on our screens every day. We also developed method to decode the different images of holograms when seen from different angles.**

I. Introduction

In this study, we are aiming to use ray tracing to simulate how optical imaging systems formulating images. This requires our skills of linear algebra, such as matrix-matrix multiplication, scalar-matrix multiplication, and knowledge of imaging, ray tracing, and properties of holograms. In this project, we also used our understanding of holograms to achieved the separation of different images from different perspectives.

II. Methods

In this project, our goal is to understanding ray tracing, create the sharp image based on the given vectors representing rays emitted from a hologram, and identify the 3 objects in the image. Ray tracing is the simulation of rays traveling in free space and in optical imaging systems. Using MATLAB, we could visualize how rays are transformed by these optical imaging systems and how they formulate clear, sharp images.

A. Ray-Tracing Through Free Space & Finite-Sized Lens

Under the instructions in part 1.4 of the case study, we simulated two sets of rays going through a thin lens and converging at the other side. The lens is place at $z = 0.2\,m$. The initial position of the rays is at $x = 0\,m$ and $x = 0.01\,m$ respectively. The three matrices, rays_in, rays_out, and rays_through_lens, represents the locations in dimension x of the rays at $z = 0\,m$, $z = 0.2\,m$, and $z = 0.4\,m$ respectively. The focal length $f = 0.15\,m$, radius $lens_r = 0.02\,m$, distance $d_1 = 0.2\,m$, and $d_2 = 0.2\,m$ so that the light emitted from a single can be converge back to a single point, creating a sharp image. The relationship between d_1, d_2, and f is shown in the following:

$$\frac{1}{d_1} + \frac{1}{d_2} = \frac{1}{f} \tag{1}$$

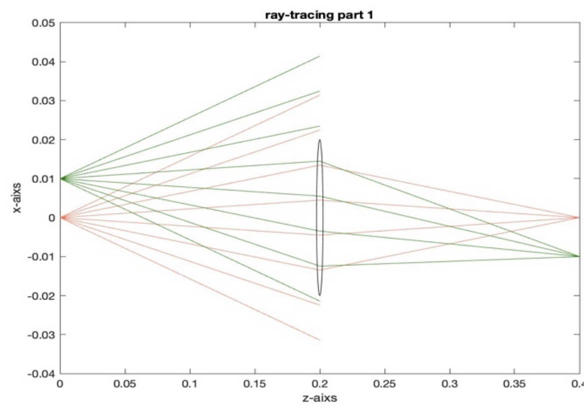

Fig. 1. Ray-Tracing through a thin lens

As shown in figure 1, the rays starting at $x = 0\,m$ reconverges at $x = 0$ m, and the rays starting at $x = 0.01\,m$ reconverges at $x = -0.01$ m, which is essentially reflected along the z-axis. This graph verified that as long as d_1, d_2, and f follows the relation of equation (1), a clear image would be displayed.

B. Ray-Transfer Matrix

We initialized distance d_1 and focal length f with arbitrary numbers $0.2\,m$ and $0.15\,m$, and d_2 based on a transformation of equation 1.

$$d_2 = \frac{f \times d_1}{d_1 - f} \tag{2}$$

Then we set up the ray-transfer matrix M_{d1}, M_f, and M_{d2} as instructed in part 1.

$$M_{d1} = \begin{bmatrix} 1 & d_1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

$$M_f = \begin{bmatrix} 1 & 0 & 0 & 0 \\ -1/f & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1/f & 1 \end{bmatrix} \tag{4}$$

$$M_{d2} = \begin{bmatrix} 1 & d_2 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5}$$

After creating the ray-transfer matrix, we use matrix-matrix multiplication to update the ray vector after travelling through some distance in the air or through a lens. R_1 represents the rays emitted from the object, and R_2 represents the rays matrix after travelling through the air and the lens.

$$R_2 = M_{d2}\, M_f\, M_{d1}\, \mathrm{R}_1 \tag{6}$$

However, in the given dataset in lightField.mat, we observed that the image created with a given function rays2img (that displays the image in grey scale) is a bit blurry. We assumed that the rays already traveled some distance before it reaches the simulate optics. The rays emitted from one point is diverged to different locations and fails to form an clear image. In other word, the transformation of M_{d1} is already performed. Therefore, we use an updated version of Equation (6) to transform the given rays in this project. R_1' is the given rays vectors; R_2' represents the transformed rays matrix.

$$R_2' = M_{d2}\, M_f\;\; \mathrm{R}_1' \tag{6}$$

We experimented with the value for d_1 and f to the values that we believe created the sharpest image.

C. Sensor Width & Number of Pixels

Initially, we set sensor width to 5 mm and number of pixels to 200 as suggested. As shown in figure 2, we tried to increase or decrease the sensor width to sharpen the image, but the image failed to get clearer because that only controls the width of the image sensor, corresponding to how large the image is displayed in the figure. Similarly, increasing or decreasing the number of pixels also did not change the sharpness of the image because it only changes the number of pixels there are. The more pixels there are, the same amount of light would need to spread to more pixels, which would result in a dimmer image, and vice versa. However, none of these methods can solve the problem of diverged rays, and therefore failed to create a sharp image.

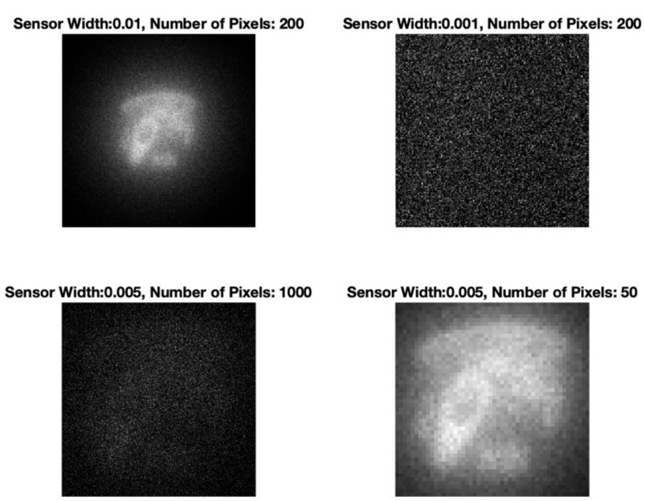

Fig. 2. The same image with different sensor width and number of pixels

D. Creating the Sharp Image

In addition to changing sensor width and pixel numbers, we propagated the rays using M_{d1} by d_1 distance greater than zero, but this would only result in an image even more blurry. The original blurry image has already been propagated in free space through a distance $d > 0$, if we propagate it again, the rays will always diverge to an even further as long as distance is greater than 0.

As a result, using the imaging concepts used from part 1, we propagated the image first through M_f, which represents the lens, and M_{d2}, which is the free space propagation after the lens. The two matrices remain the same from equation 4 and 5.

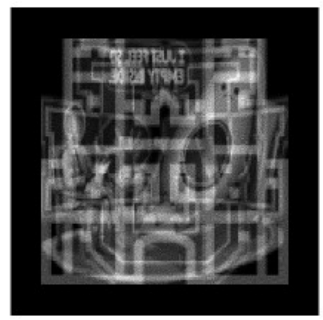

Fig. 3. Sharp Image from propagation through lens M_f and free space M_{d2}

E. Analyzing Angles with Histograms

From figure 3, we could not clearly identify the object because there are 3 objects stacked on top of each other. To separate the 3 objects, we believed the 3 objects are not on top of each other if the distance after the lens is longer or shorter than the length to focus the image. To prove that idea, we tried a slightly smaller d_2 and indeed see the 3 objects.

Based on the properties of holograms, we know that it can look different depending on the direction we are looking at it. This means that the rays propagating to different directions are not the same, and in each direction it can form a unique image. Therefore, we tried to separate the given 30,000,000 rays based on their θ_x *and* θ_y angles, and perform ray-transfer matrix independently and form images independently. The separation is achieved by plotting the histogram of the angles θ_x and θ_y, and separate the rays by the clusters of angles formed in the histogram.

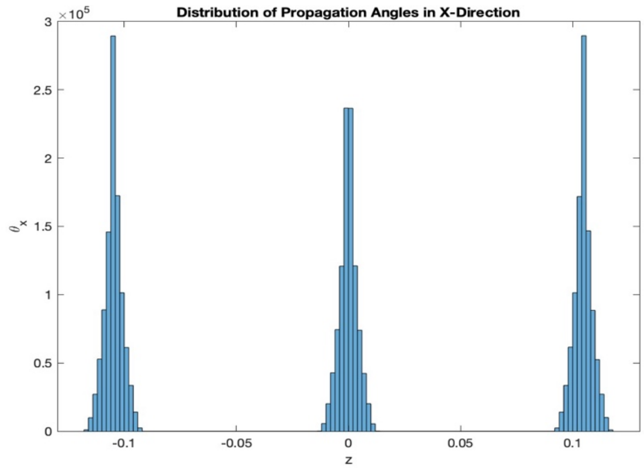

Fig. 4. Distribution of Propagation Angles in X-Direction

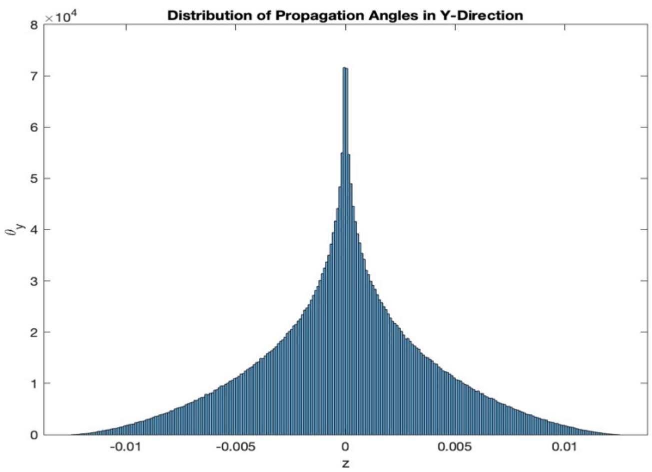

Fig. 5. Distribution of Propagation Angles in Y-Direction

Figure 5 is an expected distribution for the angles in the y direction. The single clustering of θ_y means that the image difference does not happen in the y direction. Looking at figure 4, we see that the value of θ_x has three sets of values cluttering, meaning the angles of x have three different values, which corresponds perfectly to the 3 objects we are trying to identify. We also see that there are 3,000,000 columns in rays matrix, with potentially every 1,000,000 of them representing the rays of an object. As a result, we split the rays to three based on the observed three sets of x from figure 4: $-2 < z < -0.05$, $-0.05 < z < 0.05$, and $0.05 < z < 2$. In addition, the images are mirrored due to the lens, so we multiplied rays_x by -1 to flip the image back.

III. Results and Discussion

A. Results

Figure 6 shows a logo of WashU. Figure 7 shows an avocado saying, "I just feel so empty inside" to another person who does not have a face. Figure 8 shows a 2D representation of Brookings Hall.

Fig. 6. Object 1 – WashU Logo

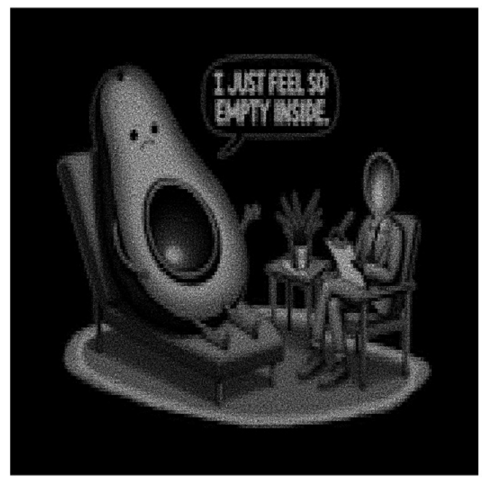

Fig. 7. Object 2 – An avocado saying "I just feel so empty inside." To another person without a face.

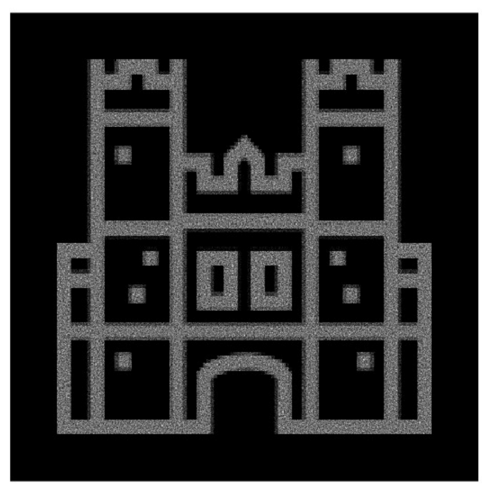

Fig. 8. Object 3 – Brookings Hall

B. Analysis of Focusing Image

In the end we set $d_1 = 0.4\,m$, $f = 0.2\,m$, and $d_2 = 0.2\,m$. This is done through a number of trials and errors. The image displayed is only 200 pixels, which is relatively small. However, increasing the number of pixels would make the image dimmer, which results in a worse image.

C. Analysis of Separating 3 Objects

Through plotting the histograms of the distribution of θ_x and θ_y, we found the three sets of theta that corresponds to the three objects in the hologram. The resulting images are relatively clear and sharp.

D. Limitations

If we were given another blurred image with a different d_1, we might not get the same sharp images with the same distance of free space d_2 and focus length f since they are set arbitrarily. The variable d_1 and f, which represents the object distance and the focus length, has to be manually adjusted. Future projects can look into the algorithm finding the best fit d_1 and f.

IV. Conclusion

A. Summary

In conclusion, by propagating the blurred image through a lens and free space, we could get a sharp image with 3 objects overlapping each other. Through separating the original rays data to three with the ranges determined from θ_x, we could separate the 3 objects and show the image clearly.

B. Future Plan

In our project, our d_1, d_2, and f are found through trial and error. In the future, we could implement a function that automatically tries different combination of values for these variables and find the optimal set that results in the sharpest image.

References

[1] S. Boyd and L. Vandenberghe, *Introduction to applied linear algebra: Vectors, Matrices, and Least Squares*. Cambridge University Press, 2018. doi: 10.1017/9781108583664.

[2] Thomas G. Smith, “Introduction to Holography,” *Internet Archive*, 1972. https://archive.org/details/IntroductionToHolography (accessed Nov. 21, 2023).