Optical Imaging as a Linear System

Sicheng Yang
Mckelvey School of Engineering
Washington University in St. Louis
St. Louis, Missouri
sicheng@wustl.edu

Mingkang You
Mckelvey School of Engineering
Washington University in St. Louis
St. Louis, Missouri
mingkang@wustl.edu

Hannah King
Department of Mathematics
Washington University in St. Louis
St. Louis, Missouri
h.l.king@wustl.edu

I. ABSTRACTION

This case study is intended to investigate how rays propagate in the real world and the methods of generating real-world digital images. There are 3 parts in the case study which are as follows: an exploration of how rays are propagated in free space by using 3D ray-transfer matrix M_d, propagation with thin lenses, and computation of an image with the inverse of M_d. In the end, our team found an optimal 3D ray-transfer matrix where d is -1 and computed a sharp and clear image of Saturn. We believe that with further research on the approach to find the optimal 3D ray-transfer matrix, this approach could be used in generating real-world camera photos.

II. INTRODUCTION

During this case study, we analyzed optical imaging as a linear system. The first part was to simulate rays leaving an object, then analyze the real-world data to explore how the rays travel through space, and how the camera sensor visualizes light. In the beginning, we defined what a sharply focused image is. Then, we explored whether changing the width of the sensor or the number of pixels would provide a sharper image. After that, we used the 3D ray-transfer matrix M_d to propagate the rays with different positive travel distances to dig into how the camera works. The second part was to explore how lenses bend rays and form images. This is equivalent to free space propagation, traveling through a focal lens, and free propagation consecutively. An image based on this system was also created to see whether our system could be used in camera sensors. The third part was to digitally refocus an image without using a physical lens as an application of computational imaging. We used the inverted propagation matrix M_{-d} with the d equal to 1 to recover a sharp image. Based on the image we recovered, we think the object is a planet with a star ring like Saturn.

III. METHODS

A. Propagation Simulation and Image From the Light-Field Dataset

For the explore ray tracing part, we first chose the distance to be 0.0001m. Next, we created the 3D ray-transfer matrix M_d ([1 d 0 0; 0 1 0 0; 0 0 1 d; 0 0 0 1]) that represents propagation in free space. In M_d, d represents the distance the rays travel through. For both of the rays in input and output planes, the initial value of the x, y, and θ_y are 0, 0, and 0. Meanwhile, the 5 values of θ_x were equally distributed between $-\pi/20$ to $\pi/20$. Our simulation was based on equation (1):

$$[x_{n+1};\ \theta_{x_{n+1}};\ y_{n+1};\ \theta_{y_{n+1}}] = M_d[x_n;\ \theta_{x_n};\ y_n;\ \theta_{y_n}] \quad (1)$$

For the part to plot the Lightfield dataset, we chose the initial width to be 0.005 m, and Npixels to be 200; while the lightField.mat's matrix recorded the ray's x positions at the first row and the ray's y position at the third row. Then, the function 'rays2img()' given in the handout was used to generate the image, and 'imshow()' was used to make the plot. We explored how increasing / decreasing the sensor width and number of sensor pixels would affect the sharpness of the image. We also explored the relationship between the image's sharpness and the distance d by using the equation (1) shown above with different d values in M_d. These relationships are elaborated upon in the discussion.

B. Exploring Lenses System and Create Image

This section was divided into three parts. The rays first experienced free propagation with $d = 0.3m$; second, the ray passed through a lens with a focal length $f = 0.15m$. Third, the rays will experience free propagation with $d = 0.3m$ again.

After the model was set up, this section was intended to research the optimal d and f to get a sharp image. After several random trials, we selected the parameter that offered us the best image quality. However, because the import data had already experienced free propagation, data were processed as M_d2 * M_f * rays.

C. Inverse Matrix to Compute a Sharp Image

First, the inverse matrix was calculated. The overall algorithm is pretty similar to the second question of part 1. However, the getM_d() parameter was set to negative numbers to receive the inverse matrix. Our goal was to try a different d and see if the image is sharper.

D. Functions

3 functions, *get_rays_track(), get_M_f(),* and *getM_d()*, were written because we observed some duplicated parts in the code. Therefore, these two functions were set up to make the code more concise. For users who want to run all implementations, please put two files in the same folder with other codes.

IV. DISCUSSION

A. Variables that Influence the Sharpness of the Images

After using the function ‘rays2img()’ with a sensor width equal to 5mm, and a number of pixels equal to 200 to plot the image, we could see a white blurry ball at the center of the black background.

We could not discern the object that was generated as the edge of the object was too blurry. We defined sharper as getting a clearer image but also maintaining detailed information about the object; increasing/ decreasing the sensor width helped us to get a better image. According to figure 1, when the width was chosen to be $5*10^{-3}$ meters, the image was much sharper than the other 5 images. The total pixels of backlit also affect the sharpness of the image. According to figure 2, Npixels equals 200, the image was sharper than the other 5 images. As a result, changing the width and total pixels could help get a sharper image. However, both solutions could not offer us perfect images.

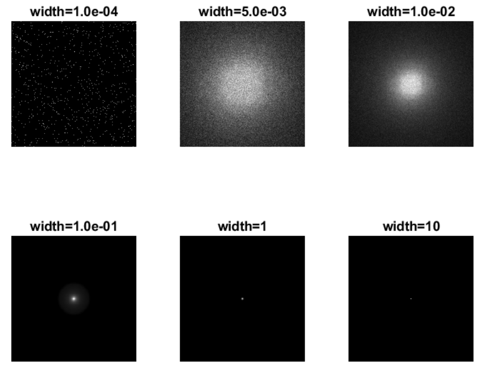

Fig. 1. Different Sensor Width Values.

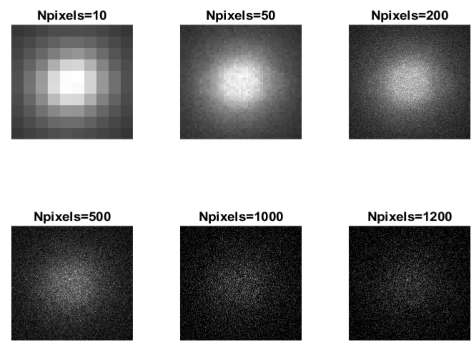

Fig. 2. Different Sensor Pixels Number.

After discussion, there were two reasons that the images were always blurry. First, the lens had a limited ability to gather rays. Second, the information would not change with different widths and total pixels.

Because the d values are always positive, the rays will never gather. The smallest possible d could minimize the propagation but it is still not sharp enough. According to Figure 3, as d becomes larger, the images are less sharp. The width and total pixels of backlit only affect how much information is received. If the width of the background is not big enough, we cannot receive the full image. However, if the width is a lot larger than required, the image will be too small compared to the whole screen. Figure 1 indicates that our guessing is reasonable.

A lack of pixels will make the image blurry, but too many pixels will reduce the energy on each pixel and make images dark. As in Figure 2, when there are only 10 pixels, the image is made of noticeable squares. However, as the total pixels become really large, the image becomes really dark. In short, to receive a high-quality image, an adequate width and total pixels should be chosen.

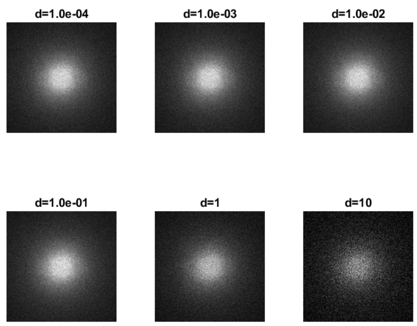

Fig. 4. Propagation with Different d Values.

B. Effectiveness of thin lens

The first section of part 2 simulated the rays which indicates that the thin lens has the ability to gather rays to a point. Therefore, after seeing the result, we believed that this lens was able to create sharp images based on the discussion in part A. Figure 5 below is our ray tracing simulation result. It is obvious that rays gathered at z = 1 at two distinct x values.

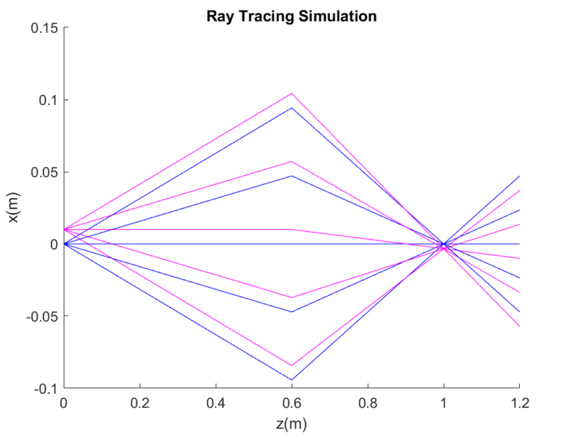

Fig. 5. Ray tracing simulation result with a thin lens

After several trials of d and f, we chose d2 equals 1, and f equals 0.5. Figure 6 below was our newly computed image. As discussed in the method, because the original data had been propagated, the first free propagation matrix was omitted.

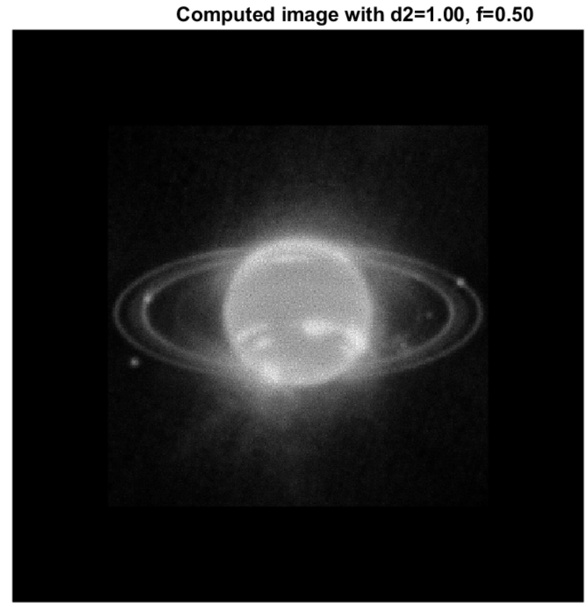

Fig. 6. Computed image with thin lens

Compared to the graph in part A, this image was much sharper. We think the object is a planet with a star ring like Saturn.

C. Effectiveness and Meaning of the Inverse Matrix

The inverse of M_d exists because M_d is a square matrix, and all rows are linearly independent. The inverse matrix is [1 -d 0 0; 0 1 0 0; 0 0 1 -d; 0 0 0 1]. The inverse matrix is identical to the propagation matrix M_{-d}.

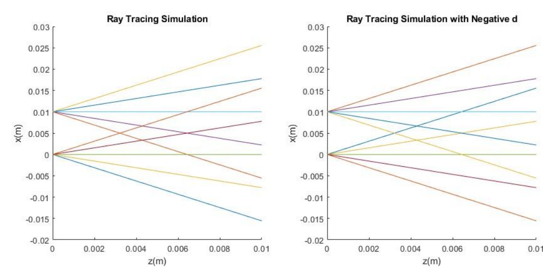

Fig. 7. Ray tracing for positive and negative d

According to the plot, for the same color rays, the directions of rays are vertically flipped. This means this material could make the light travel to a vertically inverted direction compared to material that has a positive d. After trying several d, we found the best d is -1. The image below is our computed image:

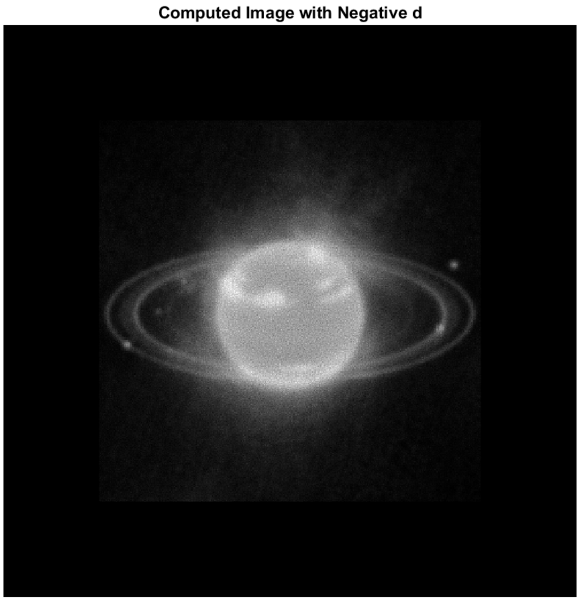

Fig. 8. The computed image

Based on our observation, we successfully compute the sharp image of the object. This image is really clear and should contain most of the information from the original image. The image has a similar quality compared to part 2. Because the lenses are different, the image is vertically flipped.

V. CONCLUSION

During this case study, we understood that a sharp image should be a focused image with a clear shape and detailed information. We found out that changing the attributes of the background where the light dots are plotted, like the width of the sensor or the number of pixels, could not provide a sharply focused image. With the understanding of how the camera works, we used the two propagation metrics M_f to simulate the rays past the lens and M_{d2} to simulate the rays passing through the space behind the lens to get a sharp focus image. We also found a negative d for the propagation matrix M_d which can gather the two sets of rays on one focused point for each corresponding ray. According to the two sharp focused images we got, we believe that the object is a planet with a star ring like Saturn.