

CS276 Homework 1: Light Field Rendering

Zhaoyu Qiu 2024192955
School of Biomedical Engineering
ShanghaiTech University

October 28, 2024

1 Introduction

The purpose of this assignment is to perform refocusing based on light field data and to experiment with different aperture effects. The input for this assignment is a 16x16 image matrix, referencing classic projects such as Lytro camera data processing.

2 Implement

2.1 Light Field Loading

The 16x16 light field input used for the text is shown in Figure 1.

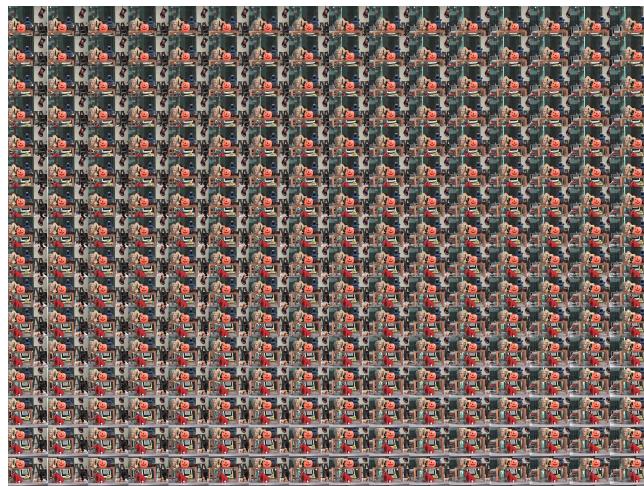


Figure 1: 16x16 Light Field Input

To create a light field, multiple images are captured from various viewpoints and organized into a 5D data structure, represented as:

$$\text{Light Field}(u, v, x, y, c)$$

where (u, v) represent the camera indices, (x, y) are the pixel coordinates, and c denotes the color channels. Each image is resized and stored in the array, ensuring uniform dimensions for accurate interpolation.

2.2 Interpolation Resampling

Interpolation is a crucial step in rendering the light field. Two primary interpolation methods are employed:

2.2.1 Bilinear Interpolation

For bilinear interpolation, the pixel value $I(x, y)$ at an arbitrary position is calculated using:

$$I(x, y) = w_a \cdot I_a + w_b \cdot I_b + w_c \cdot I_c + w_d \cdot I_d$$

where w_a, w_b, w_c, w_d are the weights based on the neighboring pixel values.

2.2.2 Trilinear Interpolation

In a 3D light field, trilinear interpolation is utilized, which extends the bilinear approach into the depth dimension. The formula is given by:

$$\begin{aligned} I(x, y, z) = & I_{000}(1-x)(1-y)(1-z) + I_{100}(x)(1-y)(1-z) \\ & + I_{010}(1-x)(y)(1-z) + I_{110}(x)(y)(1-z) \\ & + I_{001}(1-x)(1-y)(z) + I_{101}(x)(1-y)(z) \\ & + I_{011}(1-x)(y)(z) + I_{111}(x)(y)(z) \end{aligned}$$

2.3 Focal Plane Control

Let the camera position be $C = (x_c, y_c)$ and the focus distance be f . The distance from a pixel point $P = (x, y)$ to the camera position is calculated as:

$$d = \sqrt{(x - x_c)^2 + (y - y_c)^2}$$

The weight w for each camera view can be computed using the Gaussian function:

$$w = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{d^2}{2\sigma^2}}$$

where σ is related to the aperture size, influencing the weight distribution.

2.4 Aperture Control

Let A be the aperture size, which affects the weight calculation. A larger aperture allows more camera views to contribute to the final image. The relationship can be described as:

$$w \propto \frac{1}{A}$$

This means that as the aperture increases, the weight distribution becomes flatter, impacting the clarity and detail of the final image.

2.5 Z-Axis Control

The focal length factor F simulates depth along the z-axis. The displacement in the x and y directions for each camera view can be calculated as:

$$dx = (col - x_c) \cdot F \cdot \frac{\text{width}}{f}$$
$$dy = -(row - y_c) \cdot F \cdot \frac{\text{height}}{f}$$

Here, (col, row) represents the position of the camera view, and $(\text{width}, \text{height})$ are the dimensions of the final image. The blur effect in the final image is controlled by the kernel size:

$$\text{blur_kernel_size} = \max(1, \text{int}(F \cdot 2))$$

3 Experiments

3.1 GUI interface

The user interface of this assignment consists of the components shown in the Figure 2, which includes 6 control sliders, 1 display page, and 1 confirm button.

3.2 Interpolation Resampling

3.3 Focal Plane Control

3.4 Aperture Control

3.5 Z-Axis Control

4 Conclusion

This work presented a comprehensive overview of light field rendering, detailing the underlying principles and implementation techniques. The experiments highlighted the benefits of using advanced interpolation methods and the impact of camera parameters on rendering quality. Future work will explore more complex scenes and real-time rendering capabilities, further enhancing the interactive experience in light field visualization.

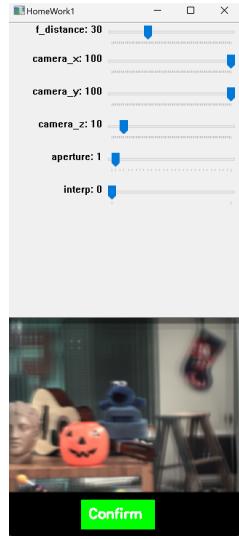


Figure 2: User Interface Overview

References

- [1] Author, A. (Year). *Title of the paper*. Journal Name, Volume(Issue), Page range.