

Assignment 1 Light Field Renderer

fanghz1@shanghaitech.edu.cn

October 8, 2020

Contents

1	Introduction	1
2	Implement	2
2.1	Data Representation	2
2.2	GUI Design	2
2.3	Render System	3
2.3.1	Moving X and Y	3
2.3.2	Disparity	4
2.3.3	Moving Z	4
2.3.4	Aperture	5
3	Experiments	7
3.1	What do you observe if you use interpolation on undersampled light field?	7
3.2	What happens when you move your focal plane from far to near?	8
3.3	Which focal plane gives you the optimal results (least aliased reconstruction)?	8
3.4	What happens when you increase the size of the aperture?	9
3.4.1	Weight change	9
3.4.2	View	9
4	Supplement	9
4.1	Implement the z directional motion of the camera	9
4.2	Making the view smoothly changed when increase the size of the aperture . .	10
5	Conclusion	10
5.1	algorithms	10
5.2	performance	11

1 Introduction

The basic light field rendering is simply realized. By adjusting the xy coordinates of the virtual camera in the st plane and the z coordinates perpendicular to the st and uv planes,

the perspective images at different positions are obtained. The simulation of the aperture is added, and the gradual change of the aperture is smoothed.

2 Implement

2.1 Data Representation

1 Input image like follow:

$$P = \left[\begin{array}{ccc} C_{11} & \cdots & \cdots \\ \vdots & \ddots & \vdots \\ \vdots & \ddots & C_{ij} \end{array} \right] \quad 240, C_{ij} = (R, G, B)$$

2 We have 256 image for the data set, for computation locality, transform the data set:

$$matrix - 2 - image(P) = \overbrace{\begin{bmatrix} R_1 & \dots & \dots & \dots & R_k \\ G_1 & \dots & \dots & \dots & G_k \\ B_1 & \dots & \dots & \dots & B_k \end{bmatrix}}^{320 \times 240}$$

$$Data = \begin{bmatrix} image - 2 - matrix(P)_1 \\ \vdots \\ \vdots \\ transofrom2matrix(P)_{256} \end{bmatrix} \quad D \text{ is a } ((256 \times 3), (320 \times 240)) \text{ matrix.}$$

3 We define two functions as using before:

Matrix 2 image and image 2 matrx Just reshape the matrix

Blend by weight Input photos like as $(wh, 3) * n$, and the weight, Output $photos * weight$ as one image.

2.2 GUI Design

1 6 scale bars, the three around the image are used to adjust the xyz coordinates of the virtual camera, and the remaining three vertical scale bars are used to adjust the disparity in the x and y directions, and adjust the size of the aperture in the aperture mode.

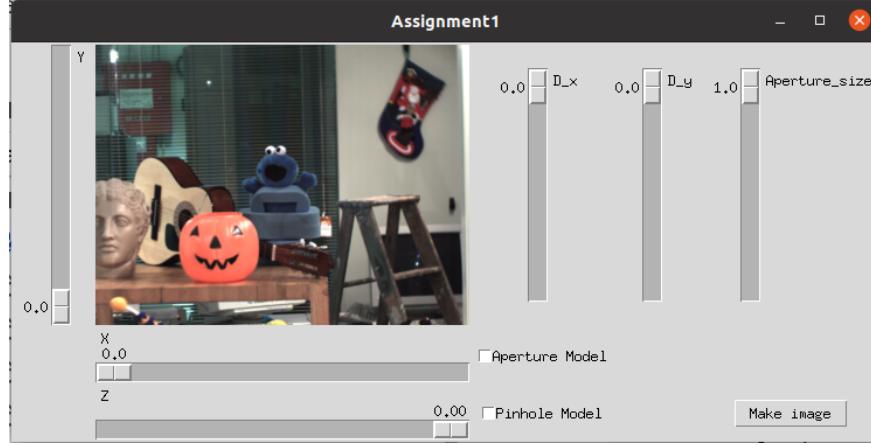


Figure 1: GUI

- 2 The adjustment range of xy is from 0 to 16, the adjustment range of z is from 0 to 1, where the zero point of (x, y) is set at the lower left corner of the st plane, and the 0 of z is set on the st plane.
- 3 The adjustment range of disparity is from -10 to 10, and the aperture size is from 1 to 10.
- 4 2 clickbox, choose if get the pinhole camera model or add add an aperture.

2.3 Render System

2.3.1 Moving X and Y

For we need getting the color blending by the nearest color,we should do the interpolation.Basic method blending from the 4 color,which have the form:

$$Camera(x, y) = [Color_1 \ Color_2 \ Color_3 \ Color_4] \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix}$$

Where the $Color_1$ have the coordinate (x_1, y_1) ,in the upper left corner of the square which include (x, y) :

$$d_1 = x - x_1, \ d_2 = y - y_1$$

$$\begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \\ \omega_4 \end{bmatrix} = Normalization\left(\begin{bmatrix} (1 - d_1)(1 - d_2) \\ d_1(1 - d_2) \\ (1 - d_1)d_2 \\ d_1d_2 \end{bmatrix} \right)$$

Basic interpolating as follow:



Figure 2: 4 points interpolation without fix

2.3.2 Disparity

The essence of Disparity is to adjust the view depth, enter the coordinates of the center position, and calculate the bias according to the surrounding coordinates.

To calculate the disparity corrected image, we simply define it as three steps:

Disparity matrix Calculate the required offset according to the center coordinates and the disparity in the xy direction

Re image Reconstruct a new picture based on the offset

Q interpolator2 Interpolate according to the new picture to obtain the required image



Figure 3: disparity(-2,2)



Figure 4: disparity(-8.3,5.6)

2.3.3 Moving Z

When the z coordinate of the virtual camera is adjusted to move away from the uv plane, the required part of the light data does not exist and the image cannot be obtained.

Therefore, it can only move to a plane close to the uv, which is similar to the image that has been obtained, keeping the center position unchanged, first performing proportional cropping, and then zooming in. Define as function scale by z.



Figure 5: $Z = 0.75$



Figure 6: $Z = 0.5$

2.3.4 Aperture

Aperture model

- 1 Think of the aperture as a circle with the input coordinates as the center and radius d , and this circle falls on the st plane.
- 2 The st plane has been gridded, and each point has a real camera image. The virtual image formed by the aperture should be synthesized using the points in the circle.
- 3 In order to make the image generated by the iris gradual, it is necessary to consider not only the points inside the circle, but also the nearest point outside the circle

Model the weight Discuss the weight of each image participating in the synthesis:

- 1 For $point_i \in all\ points\ around$ Calculate the Euclidean distance to the center coordinates $d_i \in D$ for $point_i$
- 2 For the points inside the circle, the weight of the distance part is fully included as 1. For the points outside the circle, use r/d to linearly weaken the weight, and gradually increase with r until the complete weight is obtained 1.
- 3 Regularization, so that the sum of weights is 1.
- 4 Gaussian filtering
We want to keep mainly the light close to the coordinate center, so we use Gaussian probability density function to process again.

Generate image with aperture First give a complete aperture calculation method

- 1 Input the center, also use(8.4, 8.6). Consider that the size of the aperture is 2, and then find the closest layer of points inside the aperture and outside the aperture, as shown in the figure, the green point is the center position, the red is the first layer outside the circle, and the blue is the inside of the circle points. Because all weights are not taken outside the circle. For smooth, the actual weights after mapping are green points

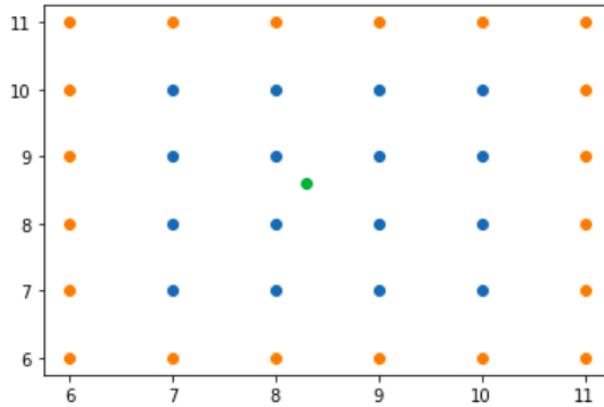


Figure 7: Points involved

- 2 According to the method mentioned above, the weights are calculated and mapped to the Gaussian distribution $scale = aperture\ size = 2$, Blue is the inner circle points, orange is the outer circle.

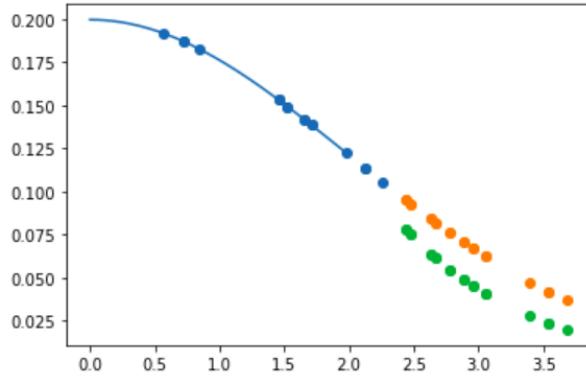


Figure 8: Gaussian mapping

- 3 In order to better reflect the focus of the weight, make the following visualization, you can see that the closer to the center of the coordinate, the greater the weight obtained.

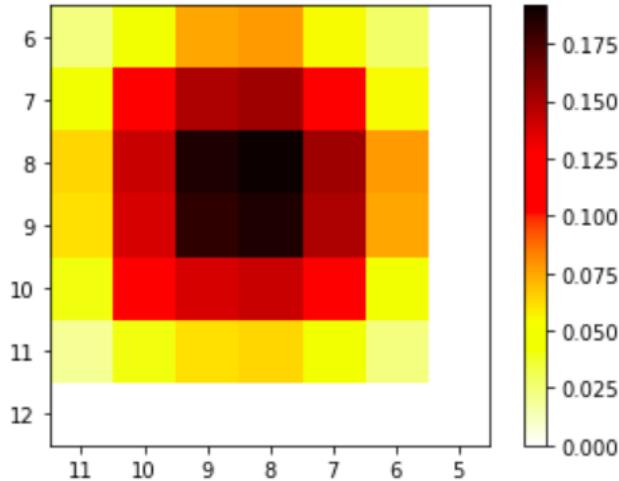


Figure 9: Weight focus

- 4 The rest of the processing methods are similar, so I won't repeat them. It can be seen that there is a more obvious blur in the background part, obtained are as follows.



Figure 10: Pinhole



Figure 11: Aperture

3 Experiments

3.1 What do you observe if you use interpolation on undersampled light field?

- 1 If the image set is undersampled, this mean we have the less information to interpolation, which would case more ghosting.
- 2 As an experiment, let's cut the image set from 16×16 into 8×8 in the same view range, than sample the $(9, 9)$ and $(4.5, 4.5)$, which mean the same direction.



Figure 12: From 16×16

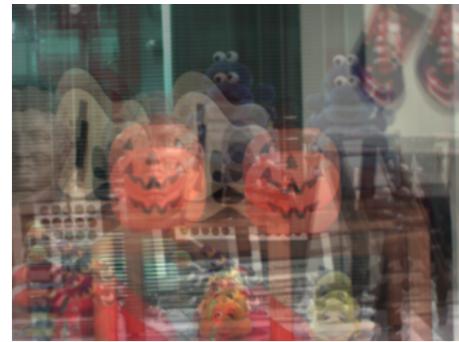


Figure 13: From 8×8

3.2 What happens when you move your focal plane from far to near?

- 1 Move the focal plane from far to near, is equal to adjust disparity. Showig as this two sampled from [8.4, 8.6]



Figure 14: Near



Figure 15: Far

- 2 From the basic image,move the focal to the near place, like showing in 2.3.2.The pumpkin and red doll in the front position become clearer, and the object and chair in the upper right corner become more blurred.
- 3 As for move to far,the scenery in the distance becomes clear. You can mainly see the ornaments in the upper right corner and the ladder on the right, but compared to the previous picture, the pumpkin becomes more blurred.

3.3 Which focal plane gives you the optimal results (least aliased reconstruction)?

It depends on the object you want to focus on. Such as Figure Pinhole is a well result in my opinion.

3.4 What happens when you increase the size of the aperture?

3.4.1 Weight change

Let's increase the size of the aperture to 4, at this time each pixel participates in the synthesis of 100 rays. At this time, the effect of weight mapping is as follows. The more scattered the samples are, the mutual bias will be equalized, and the position far away from the aperture will be blurred.

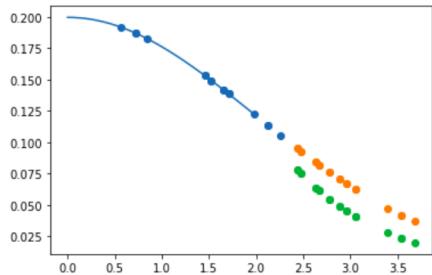


Figure 16: size=2

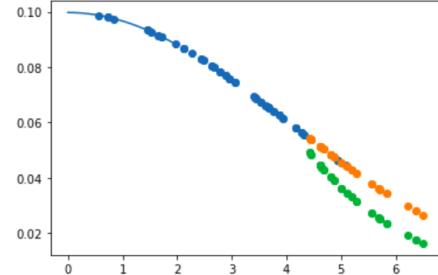


Figure 17: size=4

3.4.2 View

The blurring of the background is more obvious and darker.



Figure 18: size=2



Figure 19: size=4

4 Supplement

4.1 Implement the z directional motion of the camera

See scale by Z

$$\frac{d}{D} = Z', \text{Photo size} = (W, H)$$

$$\text{New image} = \text{Photo}[W\left(\frac{1}{2} - \frac{d}{2D}\right) : W\left(\frac{1}{2} + \frac{d}{2D}\right), H\left(\frac{1}{2} - \frac{d}{2D}\right) : H\left(\frac{1}{2} + \frac{d}{2D}\right)]$$

4.2 Making the view smoothly changed when increase the size of the aperture

See "Model the weight" part. Mainly add the nearest circle outside the circle, and then add the distance weight.

5 Conclusion

5.1 algorithms

Brightness after blending Because of the weighting and then regularization in the synthesis, physically, the larger the aperture, the brighter the scene, but the resulting composite will be darker, but directly add a plus would make overflow.



Figure 20: Lighter



Figure 21: Overflow

Currently this parameter can only be adjusted manually.

Gaussian filtering parameters The scale of the Gaussian distribution affects the weight. According to the previous conclusions, the larger the scale, the smoother it will cause the darker the background

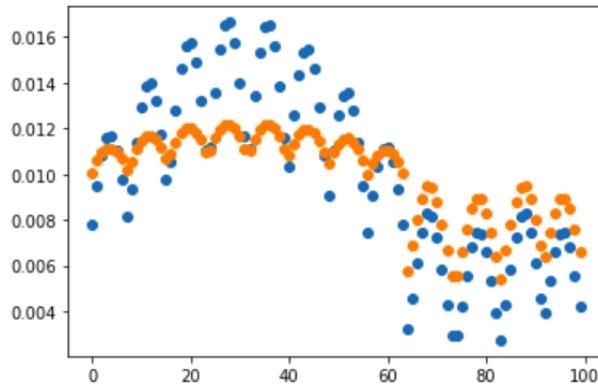


Figure 22: scale = 0.5size and size
if we add to 3size

If we add to 3size,The brightness is very low, but the blur is also obvious



Figure 23: scale=3 × size



Figure 24: Lighter

5.2 performance

Fix disparity for lots of image Numpy is difficult to control locality,I promise to use C++ implementation next time.