Computational Photography Assignment 4

Sagnik Ghosh

November 2024

1 Lightfield rendering, depth from focus, and confocal stereo

1.1 Initials

Creating the 5D array can be done using simple numpy operations:

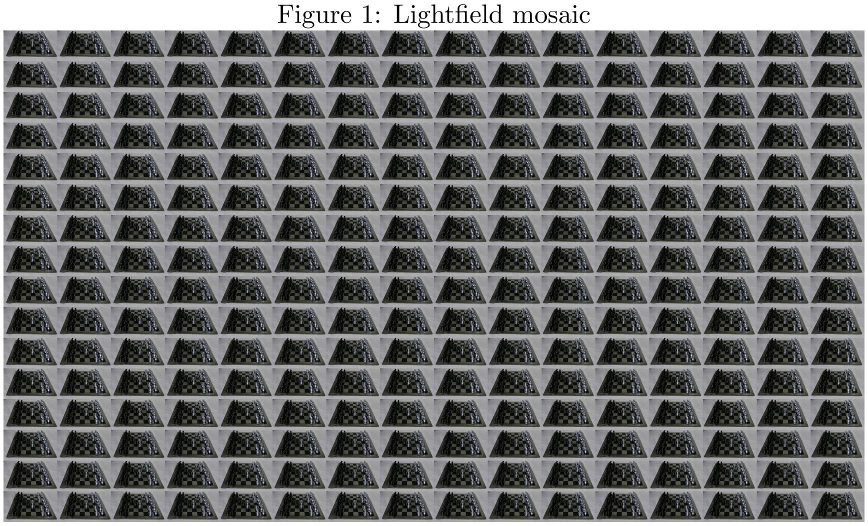
```
h, w, _ = im_lf.shape

im_5d = im_lf.reshape((h//lenslet_size, lenslet_size, w//lenslet_size, lenslet_size, 3))
im_5d = im_5d.transpose((1,3,0,2,4))
```

1.2 Subaperture views

Here's the mosaic:

Figure 1: Lightfield mosaic



1.3 Refocusing and focal-stack simulation

Figure 2: Refocused Image 1

Figure 3: Refocused Image 2

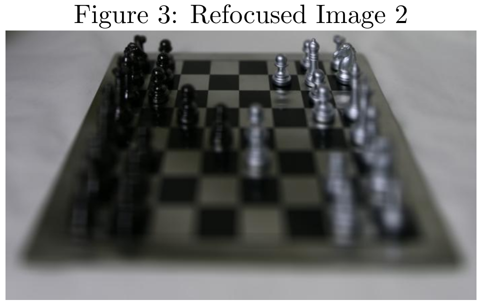


Figure 4: Refocused Image 3

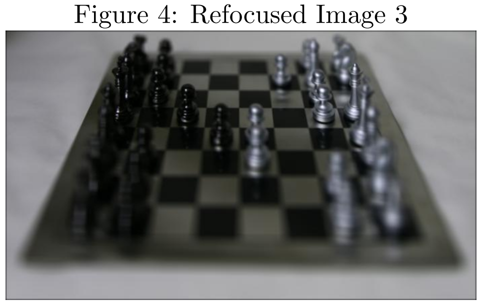
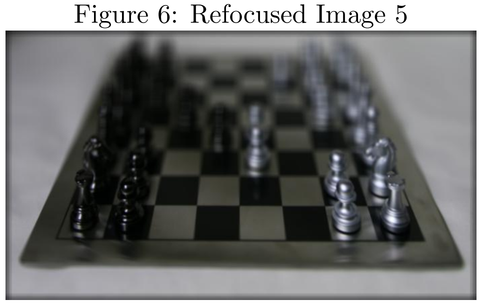


Figure 5: Refocused Image 4

Figure 6: Refocused Image 5



1.4 All-in-focus image and depth from focus

The values that worked for me are $\sigma_1 = \sigma_2 = 5$. The depth map is estimated incorrectly in regions with low texture. That's because the method relies on estimating depth by measuring sharpness ("in-focusness") of different regions and that would fail in the absence of texture. It can be seen that the depth is poorly estimated in the flat regions in between the squares. The all-in-focus image does not seem to be adversely affected by this because for regions with low texture, it does not matter if it's from an in-focus or out-of-focus image.

Figure 7: Depth from focus

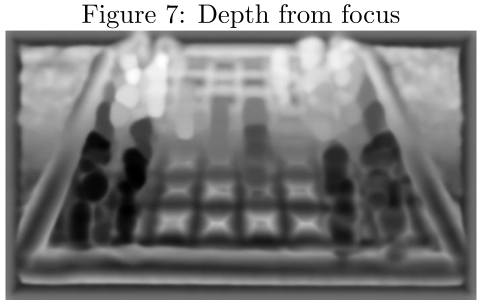
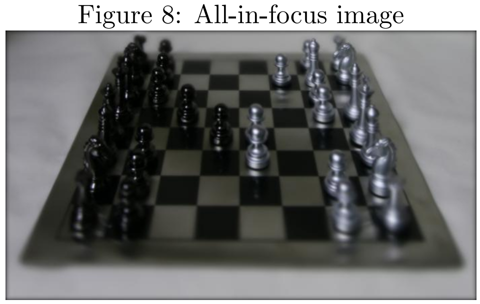


Figure 8: All-in-focus image



1.5 Focal-aperture stack and confocal stereo

The depth image recovered looks a lot noisier than the one recovered using Depth from Focus. While it does preserve more scene details, it's not a lot better than the previous depth map. Using a finer depth range would probably improve the quality, but it would be very computationally expensive to do so. It would also work better on a scene with more texture.

Figure 9: Focal-aperture stack: Aperture along vertical and focus along horizontal direction

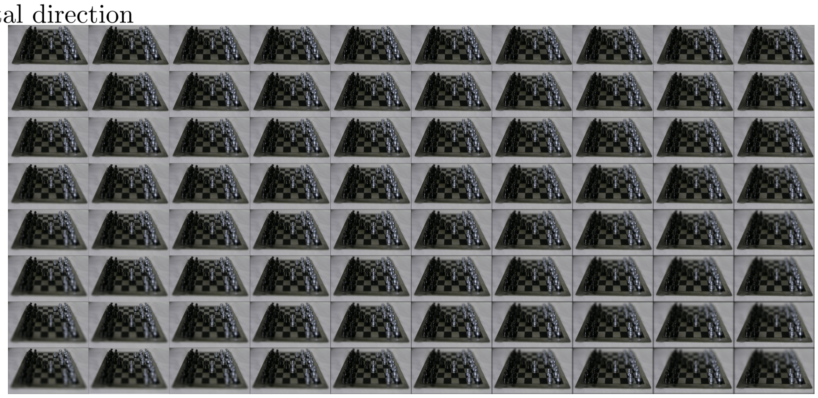


Figure 10: Depth map

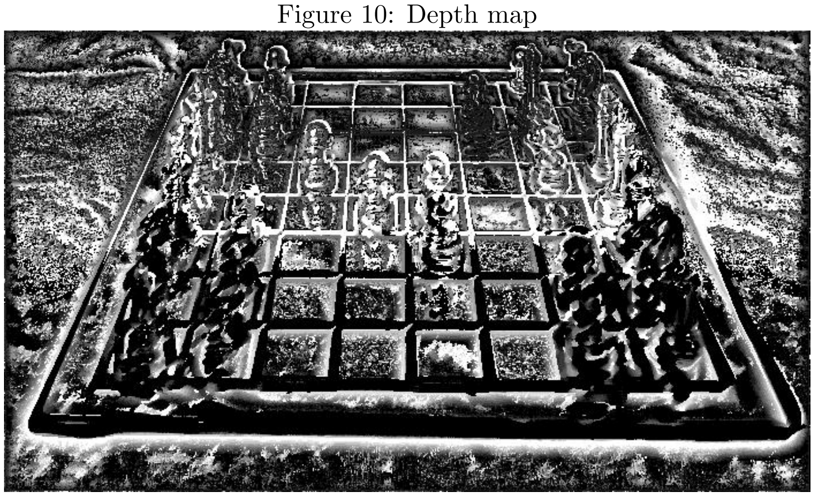


Figure 11: AFI 1

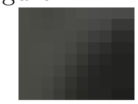
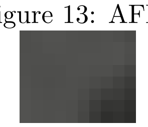


Figure 12: AFI 2

Figure 13: AFI 3



2 Capture and refocus your own lightfield

2.1 Capturing an unstructured lightfield

Here are a few frames from the video I captured. The full video is in the repository.

Figure 14: Frame 1

Figure 15: Frame 2

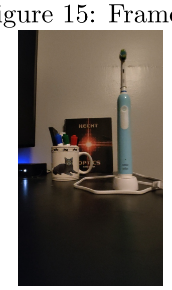


Figure 16: Frame 3

2.2 Refocusing an unstructured lightfield

To implement the normalized cross-correlation method, I split the numerator into two terms, one which corresponds to the unfiltered intensities and the other which corresponds to the box-filtered intensities, each of which can be separately implemented using the correlation function. The first term in the denominator is a constant which can be ignored. The second term has the same form as the variance of a random variable and can be expressed as the difference between the expectation of the squared variable and the square of the expectation, each of which can be implemented using convolution operations. The template I chose was the cat face on the mug. Here's the refocusing result:

Figure 17: Refocused Image