Homework Assignment 5

Due Tuesday, June 15, at 9:00 am

OVERVIEW

The purpose of this assignment is to explore light fields, focal stacks, and depth from defocus. As we discussed in class, having access to the full light field of a seen allows creating images that correspond to different viewpoint, aperture, and focus settings. We also discussed how having a focal stack of a scene allows creating an all-in-focus image and a depth map using depth from defocus.

Here, you will combine the two intro a single pipeline: Instead of capturing a focal stack, you will synthesize one from a light field image captured with a plenoptic camera. Then, from the synthetic focal stack, you will compute an all-in-focus image and a depth map.

In the first part of the homework you will be using a light field image captured by us. In the second part, you will capture and refocus a light field of your own using a standard camera.

You are strongly encouraged to read the handheld plenoptic camera paper by Ng et al. [2] that we discussed in class. Sections 3.3 and 4, in particular, will be very helpful for understanding how to solve this homework assignment. Additionally, as always, there is a \Hints and Information" section at the end of this document that is likely to help.

LIGHT FIELD RENDERING, FOCAL STACKS, AND DEPTH FROM DEFOCUS (100 POINTS)

For the first part of the homework, you will use a light field image of a chess board scene, obtained from the Stanford Light Field Archive [1].

The light field is available as ./data/chessboard lightfield.png in the homework ZIP archive. This image file is formatted in the same way as images captured by a plenoptic camera, with the pixels under neighboring lenslets corresponding to neighboring patches in the image. This format is described in detail in [2]. Figure 1 shows a crop from the center of the light_eld image, as well as a regular pinhole-camera view of the chessboard scene.

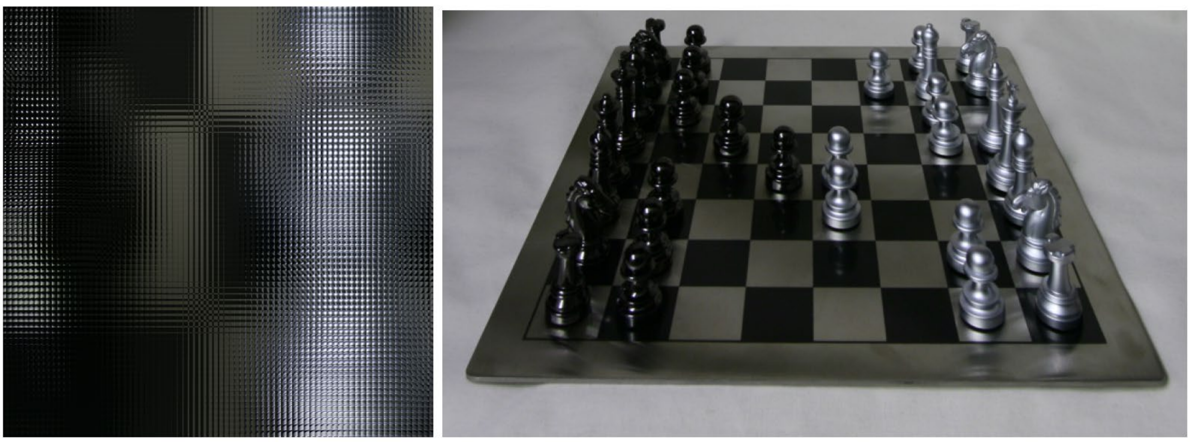

Figure 1: The chessboard scene light_eld. Left: Crop of the light_eld image. Right: A pinhole camera view of the scene.

Initials (5 points). Load the light field image in Matlab, and create from it a 5-dimensional array L(u; v; s; t; c). The first four dimensions correspond to the 4-dimensional lightield representation we discussed in class, with u and v being the coordinates on the aperture, and s and t the coordinates on the lenslet array. The fifth dimension c = 3 corresponds to the 3 color channels. When creating this structure, you can use the fact each lenslet covers a block of 16 x 16 pixels.

Sub-aperture views (20 points). As we discussed in class, by rearranging the pixels in the light field image, we can create images that correspond to views of the scene through a pinhole placed at different points on the camera aperture (a \sub-aperture"). This is equivalent to taking a slice of the light field of the form L(u = uo; v = vo; s; t; c), for some values of uo and vo corresponding to the point on the aperture where we place the pinhole. For the chessboard light_eld, we can generate 16 x 16 such images.

Create all of these sub-aperture views, and arrange them into a 2D mosaic, where the vertical dimension will correspond to increasing u values, and the horizontal dimension to increasing v values. Figure 2 shows the expected result. Submit the mosaic with your solution.

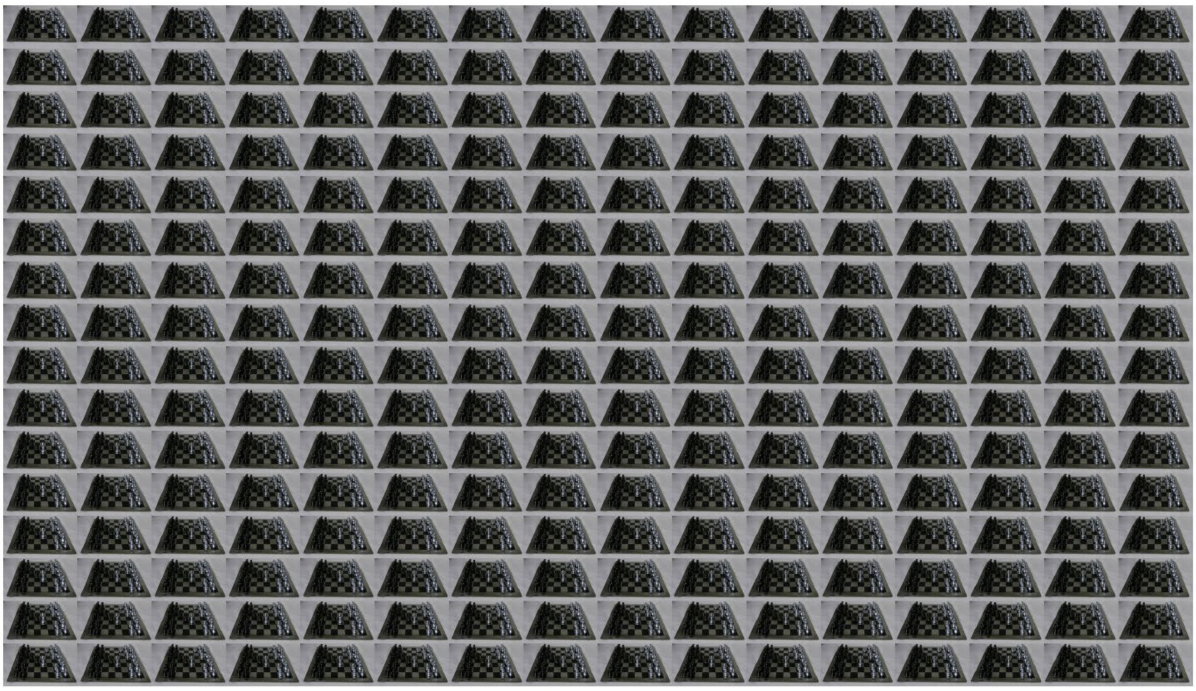

Figure 2: Mosaic of sub-aperture views.

Refocusing and focal-stack generation (40 points). A different effect that can be achieved by appropriately combining parts of the light field is refocusing at different depths. In particular, averaging all of the sub-aperture views results in an image that is focused at a depth near the top of the chess board. This corresponds to creating an image as:

$$\int_{\mathrm{u}} \int_{v} L(u, v, s, t, c) dv du. \tag{1}$$

The resulting image is shown to the left of Figure 3. As explained in detail in Section

4 of [2], focusing at different depths requires shifting the sub-aperture images before averaging them, with the shift of each image depending on the desired focus depth and the location of its sub-aperture. More concretely, to focus at depth d, we need to combine the sub-aperture images as:

$$I(s,t,c,d) = \int_{\mathrm{u}} \int_{v} L(u,v,s+du,t+dv)dvdu. \tag{2}$$

For $d = 0$, the image we obtain is the same as in Equation (1). Figure 3 shows refocused images for two more settings of d. Implement Equation (2), and use it to generate a focal stack $I(s,t,c,d)$ for a range of values d. Make sure that your focal stack is long enough so that each part of the scene is in focus in at least one image in the stack. In your solution, make sure to show at least five different refocusings.

Figure 3: Refocusing at different depths. The left image corresponds to using Equation (2) with d = 0 (or equivalently, using Equation (1)).

All-focus image and depth from defocus (35 points). As we saw in class, when we have access to a focal stack, we can merge the images into a new images where all of the scene is in focus. In the process of doing so, we also obtain depth estimates for each part of the scene, a procedure that is known as depth from defocus.
To merge the focal stack into a single all-focus image, we first need to determine per-pixel and per-image weights. This is similar to the procedure used in Homework 4 for high-dynamic range imaging, with the difference that the weights here are very different. In particular, the weights in this case correspond to how "sharp" the neighborhood of each pixel is at each image in the focal stack.
There are many possible sharpness weights. Here you will implement the following:

1. For every image in the focal stack, first convert it to the XYZ colorspace, and extract the luminance channel:
$$I_{luminance}(s,t,d) = get\_luminance\left(rgb2xyz\big(I(s,t,c,d)\big)\right) \tag{3}$$
2. Create a low frequency component by blurring it with a Gaussian kernel of standard deviation σ_1:
$$I_{low}(s,t,d) = G_{\sigma_1}(s,t) * I_{luminance}(s,t,d) \tag{4}$$
3. Compute a high frequency component by subtracting the blurry image from the original:
$$I_{high}(s,t,d) = I_{luminance}(s,t,d) - I_{low}(s,t,d) \tag{5}$$
4. Compute the sharpness weight by blurring the square of high frequency component with another Gaussian kernel of standard deviation σ_2:

$$w_{sharpness}(s,t,d) = G_{\sigma_2}(s,t) * \left(I_{high}(s,t,d)\right)^2. \tag{6}$$

Note that the weights are the same for each of the color channels.
Once you have the sharpness weights, you can compute the all-focus image as:

$$I_{all-focus}(s,t,c) = \frac{\sum_d w_{sharpness}(s,t,d) I(s,t,c,d)}{\sum_d w_{sharpness}(s,t,d)}. \tag{7}$$

In addition, you can create a per-pixel depth map by using the weights to merge depth values instead of pixel intensities, that is:

$$\mathrm{Depth(s,t)} = \frac{\sum_\mathrm{d} w_{sharpness}(s,t,d) d}{\sum_\mathrm{d} w_{sharpness}(s,t,d)}. \tag{8}$$

Figure 4 shows the all-focus image and depth map resulting from one set of σ_1 and σ_2 values used for sharpness evaluation. You should experiment with different values and report which ones work best, as well as show the corresponding all-focus image and depth map.

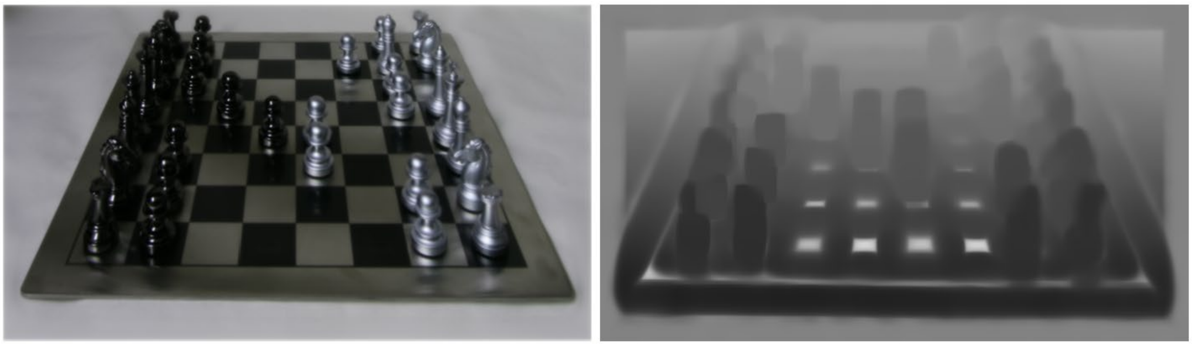

Figure 4: Left: All focus image for one set of σ_1 and σ_2 values. Right: Corresponding depth map.

HINTS AND INFORMATION

When refocusing, it will be helpful to express the coordinates u and v relative to the center of each lenslet. You can do this as follows:

```
lensletSize = 16;
maxUV = (lensletSize - 1) / 2;
u = (1:lensletSize) - 1 - maxUV;
v = (1:lensletSize) - 1 - maxUV;
```

DELIVERABLES.

For this project, and all other projects, you must do a project report in HTML. In the report you will describe your algorithm and any decisions you made to write your algorithm a particular way. Use both words and images to show us what you've done (describe in detail your algorithm parameterization for each of your results). You should submit your code and data (images), and the link for HTML by the deadline.

POLICY

Do not search the code on the web. Once detected, you will be penalized by – 200% on your project credits.

REFERENCES

[1] The (new) stanford light _eld archive, 2008. http://lightfield.stanford.edu/.

[2] R. Ng, M. Levoy, M. Br_edif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Stanford Computer Science Technical Report, 2005.