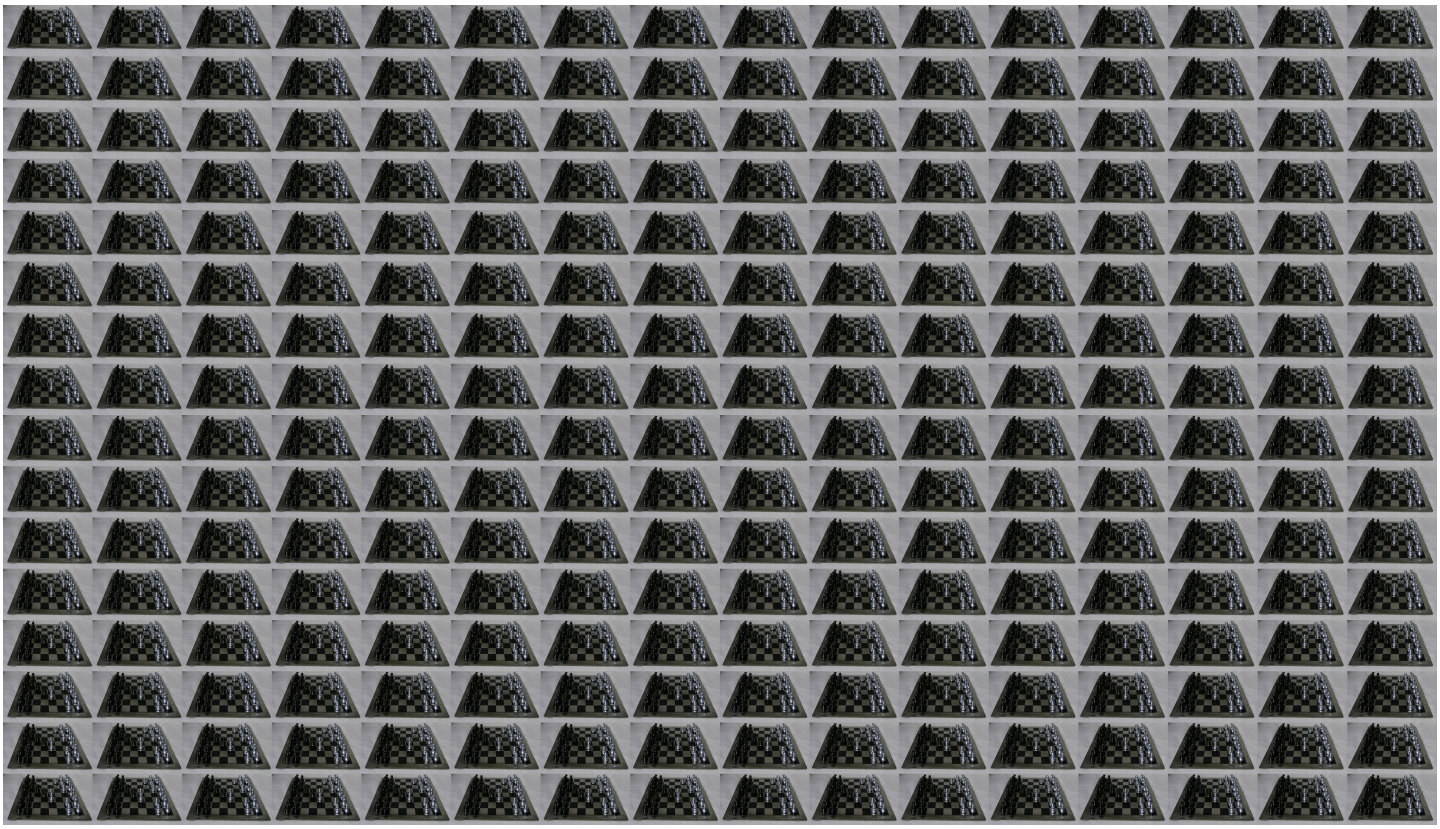
Homework Assignment 4

15-663, Computational Photography, Fall 2022, Carnegie Mellon University

Chenhao Yang

Lightfield rendering, depth from focus, and confocal stereo

Sub-aperture views



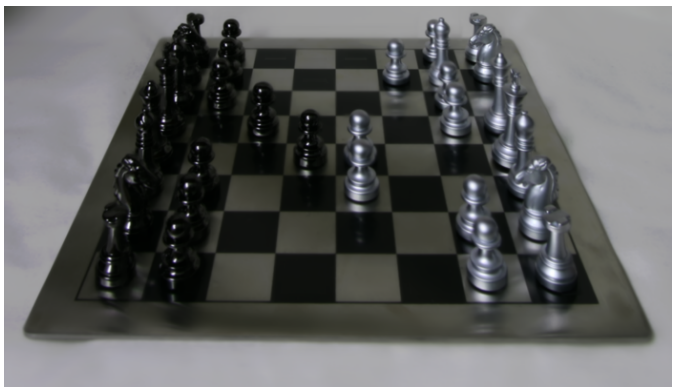
Refocusing results








All-in-focus image and depth from focus

All-in-focus	Depth from focus
	

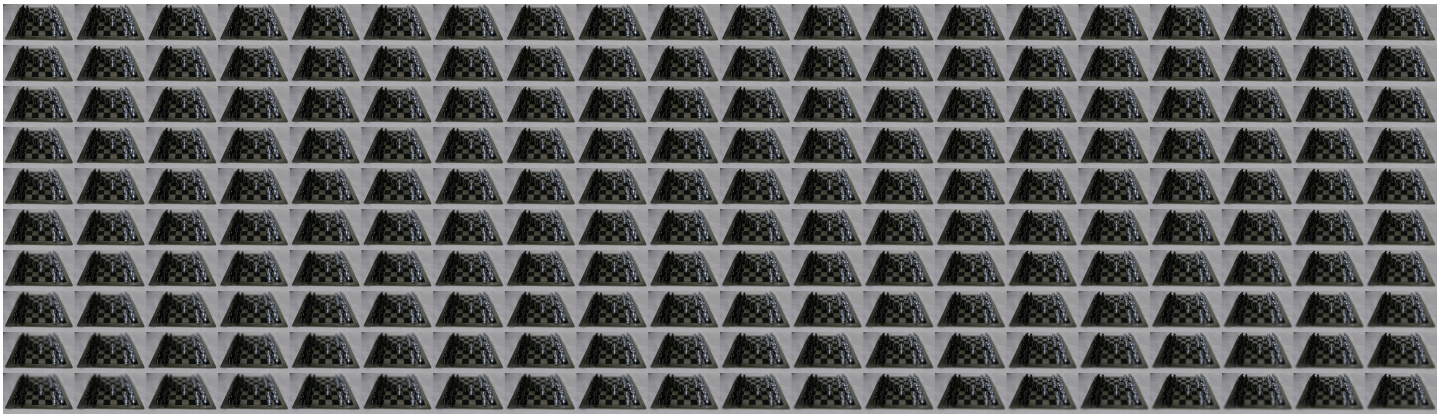
For creating the depth map, I used `kernal_size=17`, `sigma_1=0.5` and `sigma_2=2` in gaussian filtering.

The depth in parts where lack of texture such as the blank chessboard and table are estimated incorrectly. This is because depth from focus basically calculates the local sharpness of the scene and use as weights, so the scene requires rich texture to show sharpness and sharing with neighbors. In sections that lacks rich texture, like blank chessboard, there's no sharpness so the weight for these areas are close to zeros all the time.

The all-in-focus image are not affected by this issue, because pixels at these non-textured areas remain same across sub-aperture views.

Focal-aperture stack and confocal stereo

Focal-aperture stack:

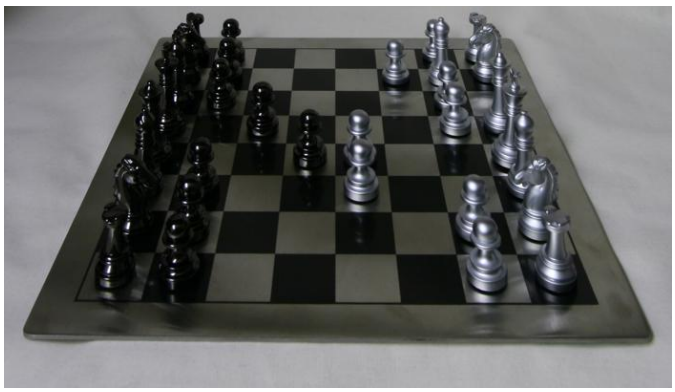
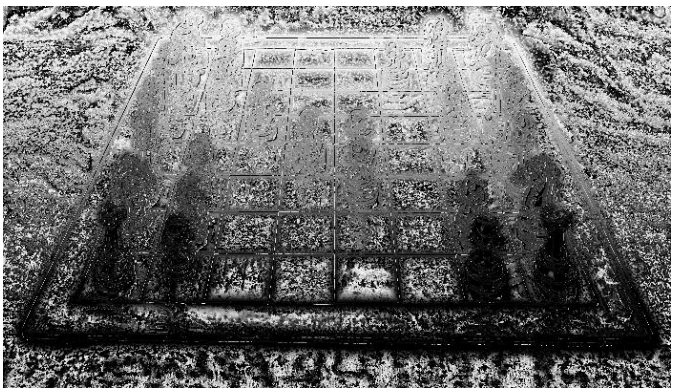


Randomly selected AFIs:



Notes: The AFIs here presented are not satisfying and we cannot visually identify too much difference across the patch. We will explain the reasons later with depth map estimated.

All-in-focus and depth map estimated using confocal stereo:

All-in-focus	**Depth map**
	

While the all-in-focus image seems fine with confocal stereo, the depth map estimated are not so satisfying compared with previous method and *Hassinoff and Kutulako*:

- We do not have a wide range of aperture and focal settings like the paper has, this make AFIs not apparent
- The scene doesn't contain same amount of texture as the paper has
- The depth map computed using confocal stereo is pixel wise and doesn't use neighbouring information like previous method, so it looks containing a lot of noise

Capture and refocus your own lightfield

The unstructured lightfield samples I captured are as follows:

The full video is located at `data/IMG_7346.MOV`.

Refocusing an unstructured lightfield

Here we are matching template from a patch of the scene by computing normalized cross-correlation. I used `scipy.signal.correlate2d` to calculate the numerator and demoninator of equation (9):

- the numerator

$$\sum_{k,l}(g[k,l]-\bar{g})(I_t[i+k,j+l]-\bar{I}_t[i,j]) = corr(g-\bar{g}, I_t) - \sum_{k,l}(g[k,l]-\bar{g})\ \bar{I}_t[i,j] \tag{1}$$

$$= corr(I_t,\ g - mean(g)) - 0 \tag{2}$$

$$= corr(I_t,\ g - mean(g)) \tag{3}$$

- the demoninator

$$\sqrt{\sum_{k,l}(g[k,l]-\bar{g})^2 \sum_{k,l}(I_t[i+k,j+l]-\bar{I}_t[i,j])^2} \tag{4}$$

$$= sqrt\left(sum(g-mean(g))^2 * \left(corr(I_t^2, box_g)^2 - 2 * corr(I_t, box_g) * corr(I_t, box_g) + corr(I_t,\ box_g)^2\right)\right) \tag{5}$$

$$= sqrt\left(sum(g-mean(g))^2 * \left(corr(I_t^2, box_g)^2 - corr(I_t, box_g)^2\right)\right) \tag{6}$$

code-wise (in `python`):

```
box_g = np.ones_like(template_g) / (template_g.shape[0] * template_g.shape[1])
h_numera = correlate2d(image_g, template_g - template_g.mean(), mode="same")
h_demoni = np.sqrt(
    np.sum((template_g - template_g.mean()) ** 2)
    * (
        correlate2d(image_g**2, box_g, mode="same") ** 2
        - correlate2d(image_g, box_g, mode="same") ** 2
    )
)
h = h_numera / h_demoni
```

Results

Focus on tiger	Focus on pumpkin	Focus on flower
		

For competition: