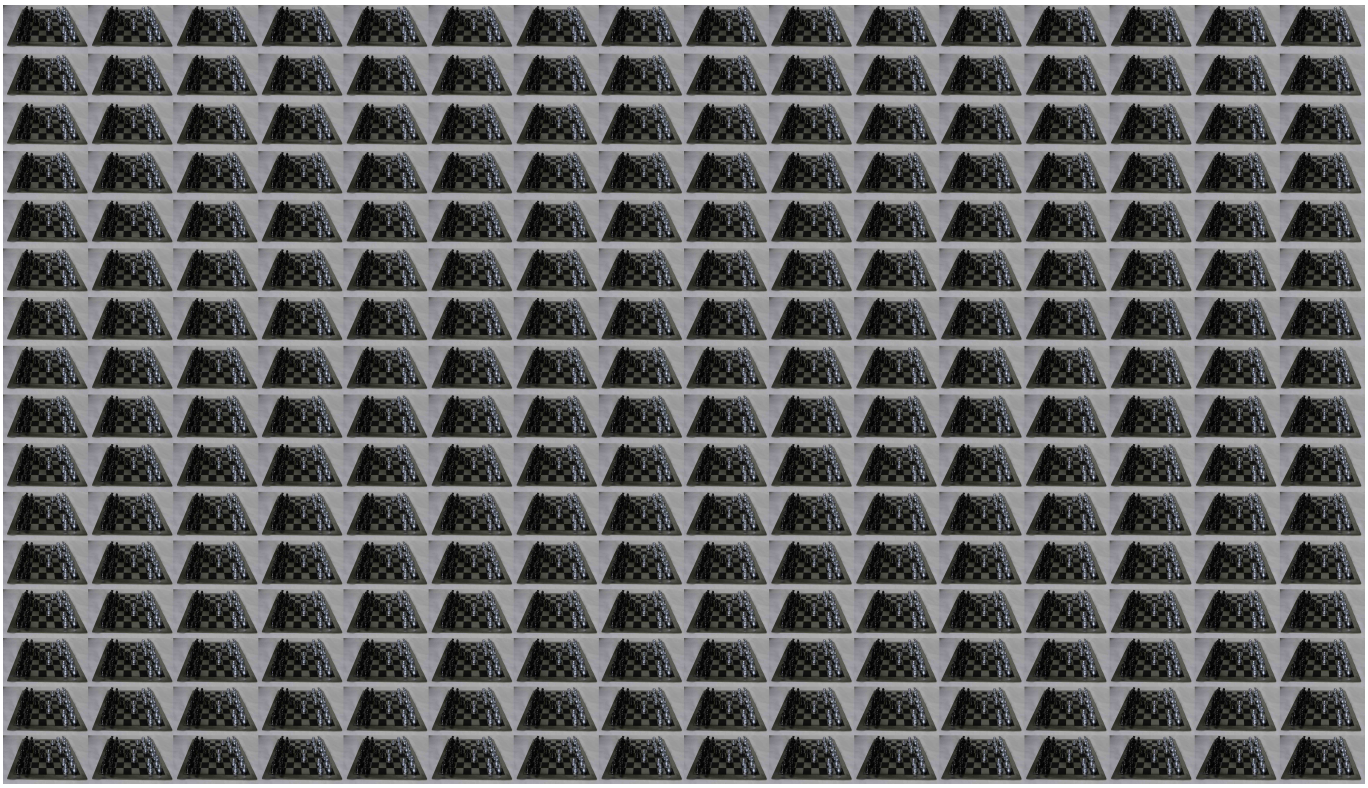
Homework Assignment 4

Instructions to Run

```
python main.py --args
--lightfield: Path to lightfield image.

--create_mosaic: If included, will create mosaic of sub-aperture views
from lightfield and write to ./mosaic.png

--get_focal_stack: If included, will simulate focal stack for depths
[-0.4, 1.9] with a step of 0.1 and will write to ./focal_stack/

--all_focus: If included, will create all focus image and write
all_in_focus_{sigma1}_{sigma2}.png and depth_{sigma1}_{sigma2}.png to
./focal_stack/

--sigma1: σ_1 to use in all-focus image

--sigma2: σ_2 to use in all-focus image

--get_focal_aperture_stack: If included, will simulate focal stack for
depths [-0.4, 1.9] with a step of 0.1 and square apertures of size [2, 16]
with a step of 2, and will write to ./focal_aperture_stack/

--get_focal_aperture_collage: If included, will create focal-aperture
collage and write to ./focal_aperture_stack/focal_aperture_collage.png

--confocal_stereo: If included, will use focal_aperture collage to recover
pixel-wise depth and will write to ./confocal_stereo_depth.png

--lightfield_video: Path to lightfield video. If included will extract
frames to ./lightfield_frames/, and will display for manual selection of
patches for template selection and determining search window bounds. Will
then refocus around template and write to
./lightfield_frames/refocused.png`
```

1. Lightfield rendering, depth from focus, and confocal stereo

Sub-aperture views (20 points)



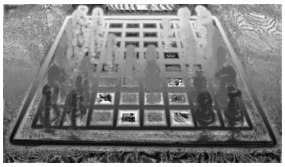
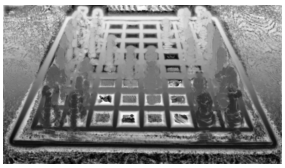
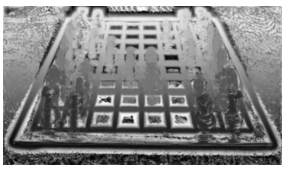
Refocusing and focal-stack simulation (25 points).

The focal-stack was simulated over the depth range [-0.4, 1.8].

Depth	-0.4	0.0	0.4	1.0	1.7
					

All-in-focus image and depth from focus (25 points)

σ1/σ2	1.0	2.0	3.0	4.0
1.0				
2.0				
3.0				
4.0				

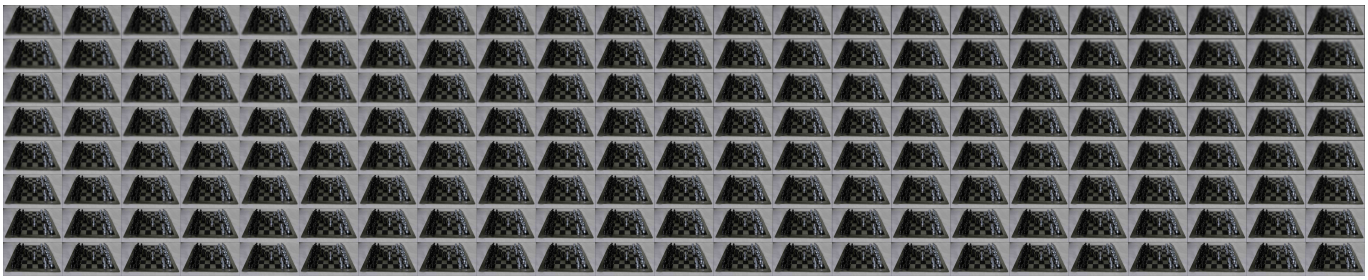
σ1/σ2	1.0	2.0	3.0	4.0
1.0				
2.0				
3.0				
4.0				

It seems as though larger σ's in general seem to yield smoother depth maps, without much visual change to the all-in-focus image. The parts on the depth map that seem to be most incorrectly estimated are those where there is a very uniform texture as these will have a very low (or zero) sharpness weight depending on

the kernel size (which was automatically selected by OpenCV in this case). The all-in-focus image is not as visibly affected by these low sharpness weights however because it is hard to visually distinguish depth changes in regions with uniform texture.

Focal-aperture stack and confocal stereo (25 points)

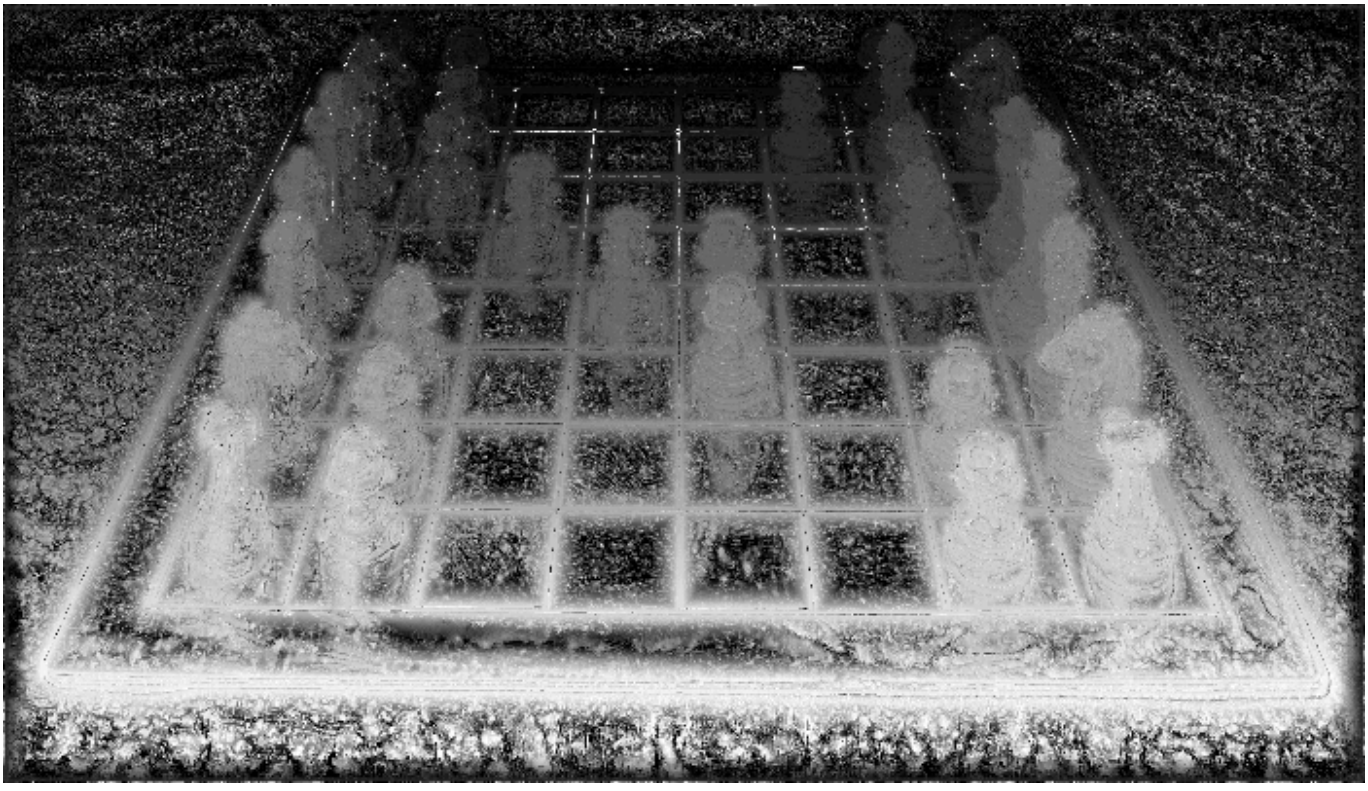
Focal Aperture Stack Collage



AFIs

Pixel Position [y, x]	[0, 200]	[100, 500]	[300, 500]
			

Depth Map



Compared to the depth map recovered from depth-from-focus, this depth map is much more noisy. However, this is to be expected because we are computing per-pixel depths. Overall, it seems as though this depth map still returns a sensible result.

3. Capture and refocus your own lightfield (100 points)

Refocusing an unstructured lightfield. (70 points)

The first assumption I made here is that the first frame in the video sequence is at the upper-left-most point in the trajectory and that the last frame in the video sequence is at the lower-right most point in the trajectory. I then take the middle frame in the sequence as my reference frame and select an image patch around which I would like to focus. Then I select the same patch in the upper-left-most frame and the bottom-right-most frame. Given these 3 patches, I can compute patch centers and the necessary size of the search window by computing horizontal and vertical distances between patch centers in both directions. This can be seen in `get_template_and_window()`. Then, for each frame in the sequence, I take the luminance channel of both the template and that frame, slide a template-size window across my search window, and at each position compute the NCC between the current patch and my template. I then extract the indices of the center of the patch that had the highest NCC (ie. was most similar to the template) and translate appropriately to compute the shift between the patch center in the middle frame and the patch center in the current frame. This can be seen in `matchTemplate()`. Once this shift has been recovered, I shift the current frame accordingly. Once all frames have been shifted, I sum all frames and take the mean to get the refocused image. This can be seen in `refocus_unstructured()`. Below are some results.

Focus Object	Rubiks Cube	Spray Paint	Cologne	Spike