Ikhlas Attarwala / Fall 2019
EECS 331 / Computational Photography
Group #9 (w/ Yunhao Li, Deanna Dimonte)

Synthesizing a Wholly-Focused Image

Couple things before I begin:

- There are 2 matlab files that I created.
    - **main.m** is the beginning of my work up until I calibrate a focal stack.
    - **formula.m** incorporates the last two parts where I compute a depth map and recover an all-focus image. I had to separate the files because if I wanted to test multiple K values, it was easier to turn this file into a formula that was fed the images, K value, and Laplacian kernel.

This week we looked at synthesizing new images that combined all the sharply focused elements of multiple images into one to remove blur from depth. Our first assignment was capturing images with our android tablet. We took 30 images with small sized objects such as batteries and bottles on tiled ground. We tried to keep the camera at the eye-level of the small objects and placed the objects behind one other. We tried to incorporate some texture to each (such as frills on a green bottle cap or wordings on the battery). After each picture, we increased the lens' focus distance so that in the next picture, a different depth would come out sharper. We recorded all of these measurements and incorporated them into our MATLAB program.

Here our first step was calculating a lens-to-sensor distance in the file main.m. We calculated each as the difference between the focal length of our rear camera and focal distance (aka. object distance) of the image. Then we proceeded to rescale each image. I saw minimal change when comparing the rescaled version and original, but do know that it caused problems later on because the original image was previously resized by me to be 640x480, and this is why I believe there's an unnecessary border in my results. It's not a major concern. Once rescaled, I wrote formula.m.

In order to compute the depth map of any image, it would be easier to convert the image to grayscale. Laplacian has been quite useful in image processing for tasks ranging from blob detection to edge detection. Laplacian gives us strong positive responses for edges, as it can show a jump or high variation in intensity in these regions. Where we have our strongest edge, that's our depth. In fact, Laplacian kernel splatting is even used in cinematic depth-of-field synthesis and reconstruction.

Our objective was to find where (which pixels) the focus was at maximum in the focal stack. One of the arguments necessary for formula.m takes in a kernel, and I used the one provided on the homework page. K was a value we used to offset the range of our focus measures. Figure 1 shows a comparison between the depth map, depth index map, and fully-focused image at different K-values. Depth map and depth index map are odd terms, but really I meant to say they were different maps at earlier and later stages in the process. Ollie's website used both terms, so I was confused for a bit.

Finally, we compute the result as a function of the pixel locations and depth map of the scene to retrieve an all-focus image. Unfortunately, I didn't get the desired results… or maybe I did? Ollie's examples and website show some haze and noise around the final picture even though in individual frames, the cat/goat/dog may be clearer. In my final result, there is a similar haze around the edges of my images. I also experimented with various K-values to see how it affected our results. Increasing the

K-value does to some degree reduce the haze when the objects are nearby, but the background remains a bit distorted no matter what. Note that I mean this for the final result. It instead appears to have a different effect on the depth map, as shown in Figure 3, and makes things much noisier and more distorted.

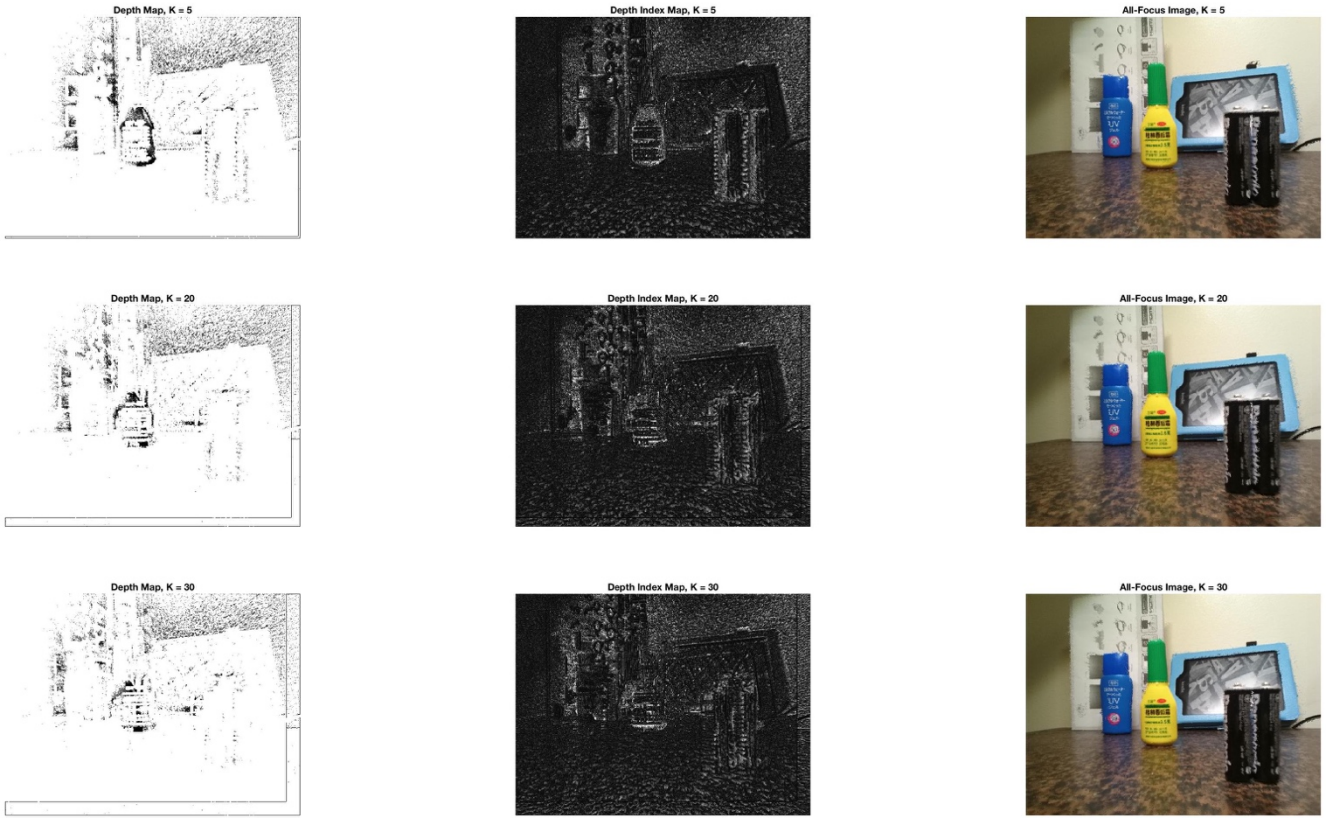

Figure 1: Comparing depth maps and the wholly-focused image. It might be easier to view in the results folder. I've also used different K-values [5, 20, 30]

An example original image

All-Focus Image, K = 50

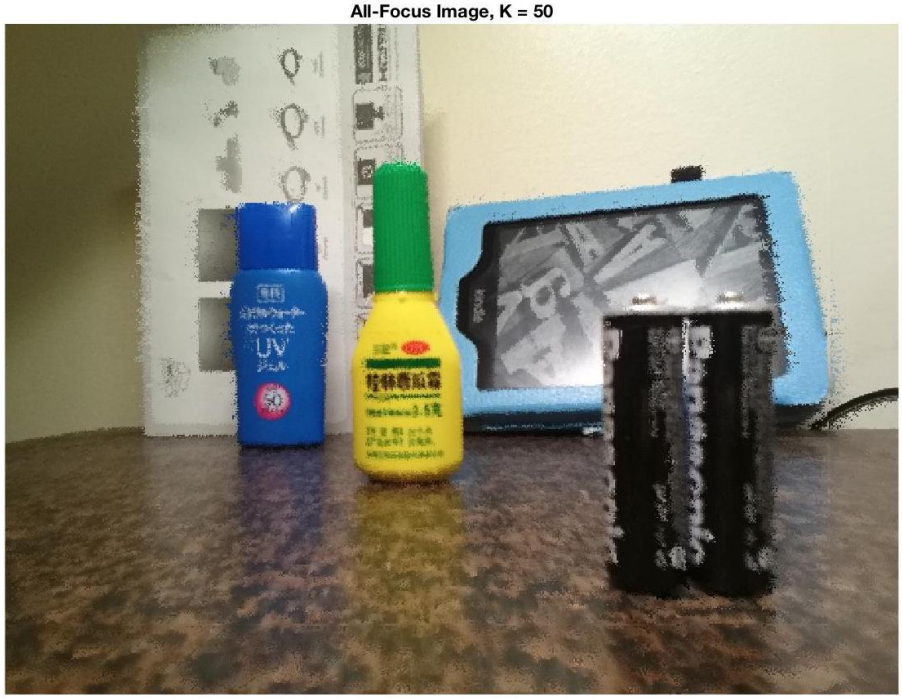

Figure 2: Comparing the original image to the final result.

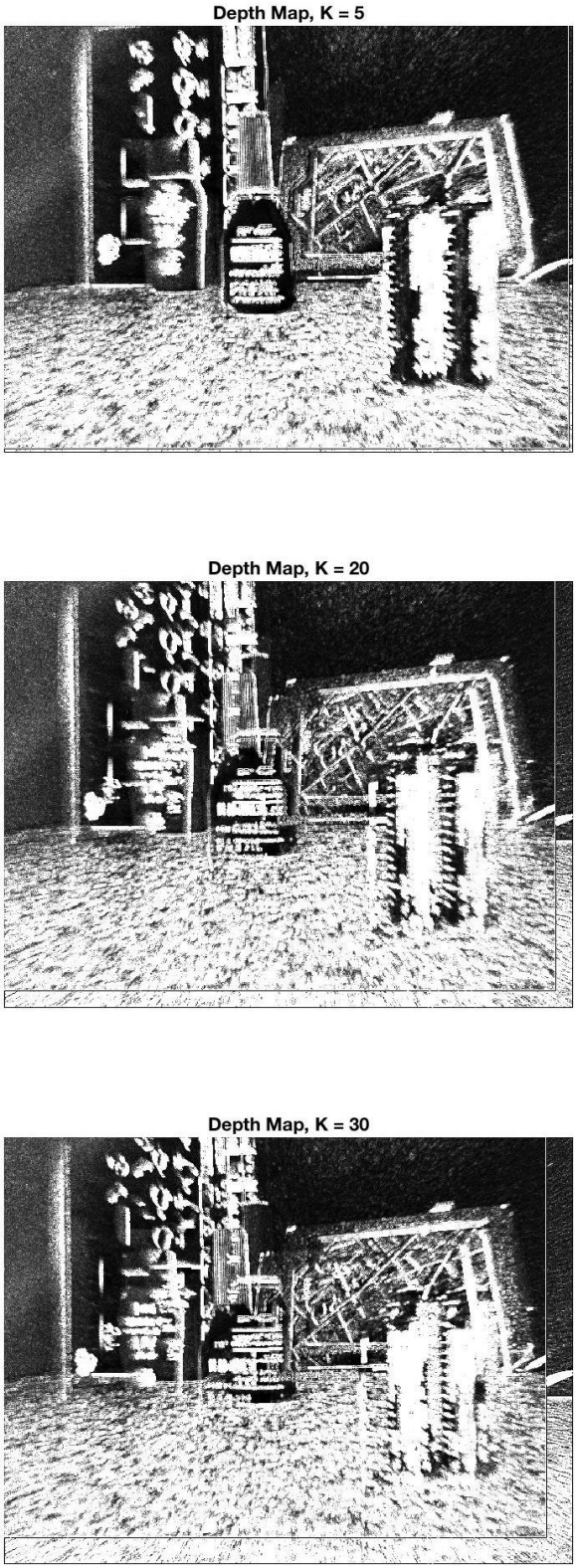

Figure 3: Comparing K values only on the Depth Map. We can see greater distortion as K increases, but that seems to not be the case in the all-focus images.