Thomas Carriero

COMP 590-175

Dr. Chakravarthula

Wednesday, April 17th

PART 1:

For scaling with darkness <black>, saturation <white>, and multipliers <r_scale> <g_scale> <b_scale> <g_scale>, the values are as follows:

<black> = 16383

<white> = 0

<r_scale> = 1.628906

<g_scale> = 1.0

<b_scale> = 1.386719

For the 2D array of unsigned integers after using the skimage imread function, the width of the image is 4284, and the height of the image is 2844. The image has 16 bits per pixel.

To identify the correct Bayer pattern, I used the Matplotlib function imsave to store a 2x2 grid of the top left corner pixel of the baby image, which resulted in the image that can be seen in the project directory under 'data/top_left_square.png' (you will have to zoom in a lot to see this). Since the top right subpixel and bottom left subpixel are the same color, then the pattern must be either BGGR or RGGB. To determine which of the following patterns was correct, I experimented with using both images in each step of the process. After demosaicing and color space correcting, it was clear that the RGGB image was closer to the real image, as the baby's shirt appeared red rather than blue, keeping consistency with the true image.

After producing images with the white world, gray world, and camera presets white balancing algorithms, I have determined that the camera presets image looks the best. The white world and gray world images are pretty similar, but the gray world might have a slight edge over the white world image because the white world image seems to have more of a green tint and grainy texture. Both of these images, however, have a green tint and appear to be too bright when compared with the real image. The camera presets image seems to be very close to the real image, and the brightness and tint seem to be extremely accurate to what was expected. This image can be seen in 'data/camera_presets/final_image/compression_wb.png' (you may have to zoom in and out once for the image pixelation to normalize).

When experimenting with values for the gamma encoding, I determined that the image appears best when setting the post-brightening grayscale mean value to the expected 0.25. Whenever I decremented the value by even 0.05, the image appeared noticeably darker. I had the same issue whenever I incremented the post-brightening mean value by any amount, but the images would appear too bright. This effect was even more apparent in the camera presets image than the white world and gray world images.

Comparing the compressed images (I rendered one with a quality of 100 and one with a quality of 95) to the original, I could not really distinguish any of them from one another since the quality seemed pretty good for all three images. The uncompressed image might be slightly better than the other two in terms of sharpness. At a compression quality of 95, the compression ratio is 22.6, and at a compression quality of 100, the compression ratio is 16.6. If the compression ratio is anything lower than 16.6, the image becomes distorted and noticeably worse than the other compressed images.

When performing manual white balancing, I found that dull white patches in the image worked best. At first, I chose the brightest white patches (the ones in the top right quadrant of the image), but these made the resulting image appear too bright. When I tried using darker patches, the image appeared way too dark, so I decided to use the dull white/gray patches in the top left quadrant during the manual white balancing, and this resulted in the best image.

When using dcraw to process the image, I used the following commands with their respective flags:

```
dcraw -v -w -q 3 -o 1 -g 2.4 12.92 data/baby.nef
convert data/baby.ppm baby_dcraw.png
```

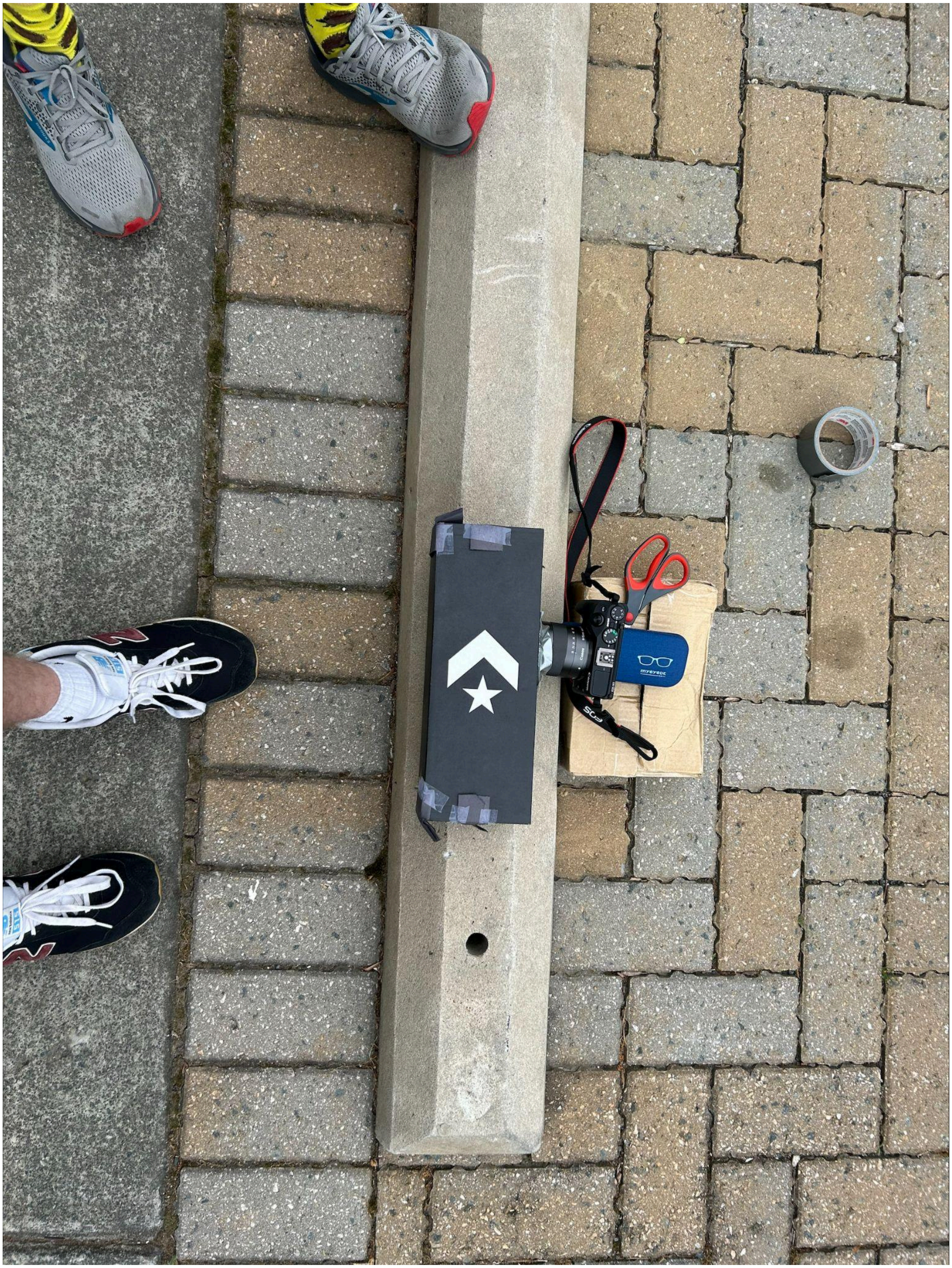
The best image was the one produced by using the dcraw command. This image seemed to match the darkness of the original image best, and it captured a lot of the cool, dark hues of the images (like the baby's eyes) better than the images produced by my code. That being said, I think my camera presets final image did a great job of converting the raw image to a finished image, but it was still slightly too bright.

PART 2:

The camera obscura I used was made of a shoebox lined with black paper on all sides except for the white screen. Any places of the box that let in light easily, such as the corners, were covered with extra tape and paper to ensure that all light was only coming through the pinhole. The focal length between the pinhole and screen was between 3.8 and 4.1 inches, and the screen was about 4 inches by 10 inches. The field of view for the images captured was anticipated to be smaller, but due to constraints on available materials, I made due with what I had and ended up with some slight blurriness in most of the images. Images were captured by connecting the camera to a larger opening that was covered in tape again when the camera was inserted. I made the first pinhole with a pin, and the other two were made by poking a hole into the black construction paper with a pencil lead.

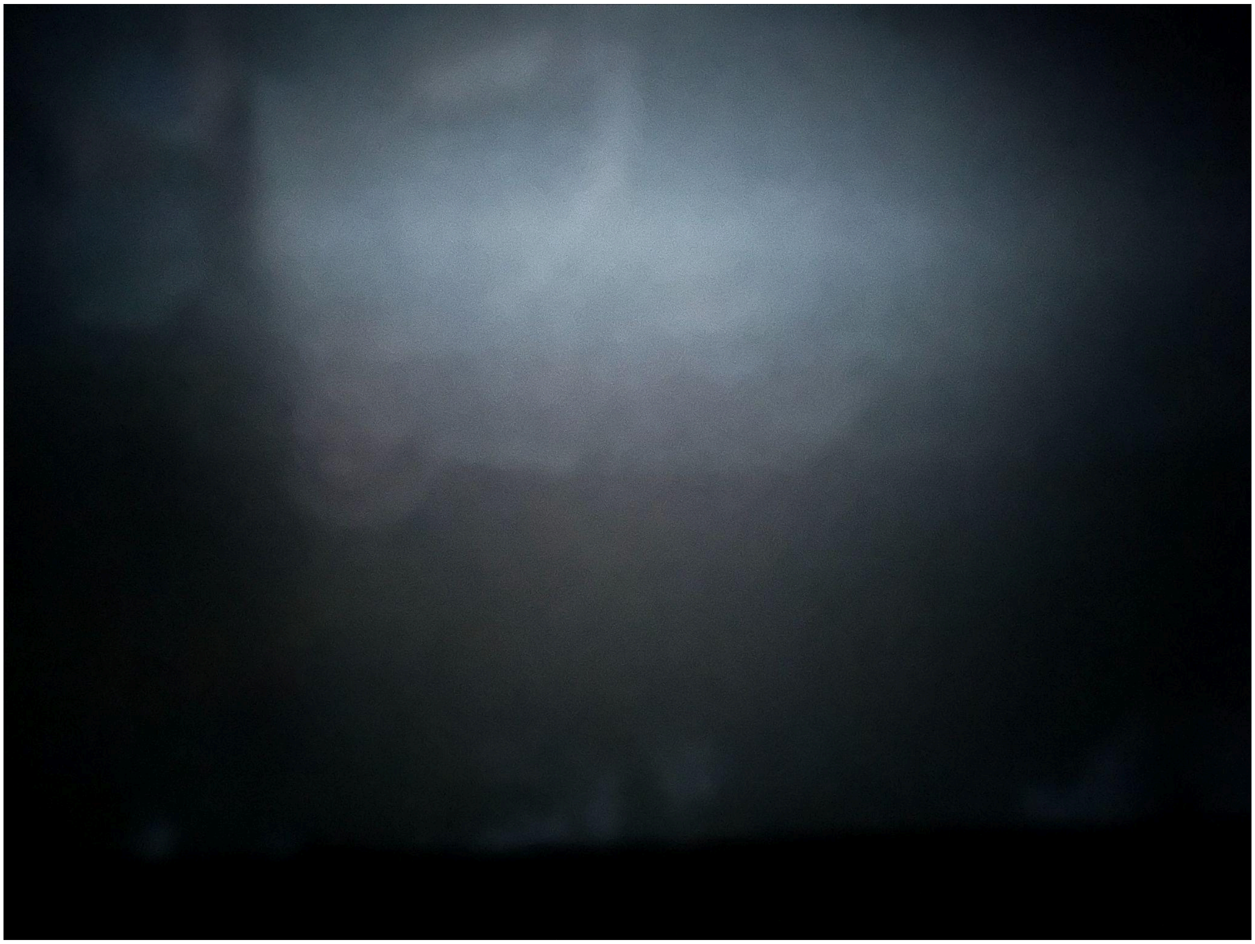

CONVERSE

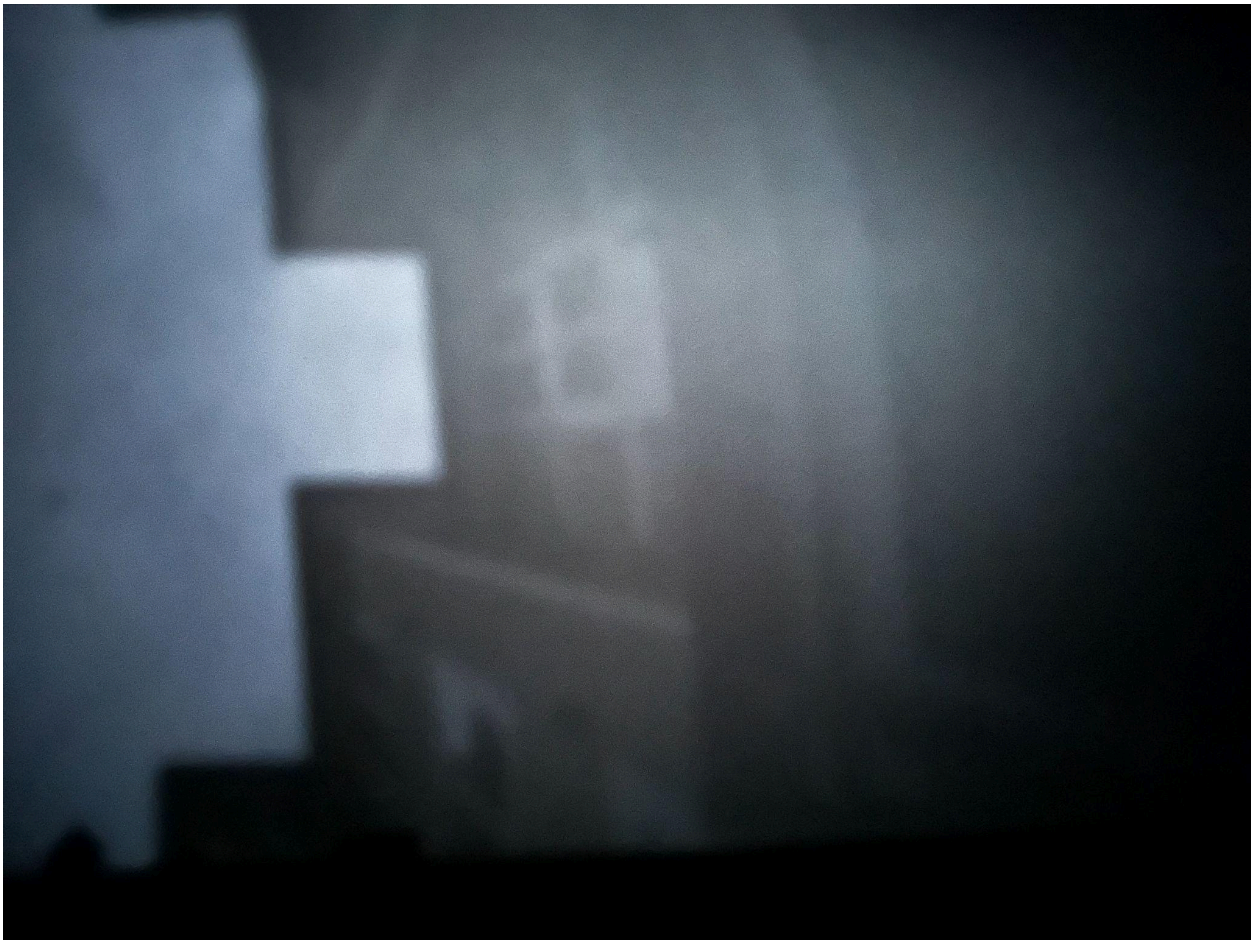

CONVERSE
3M

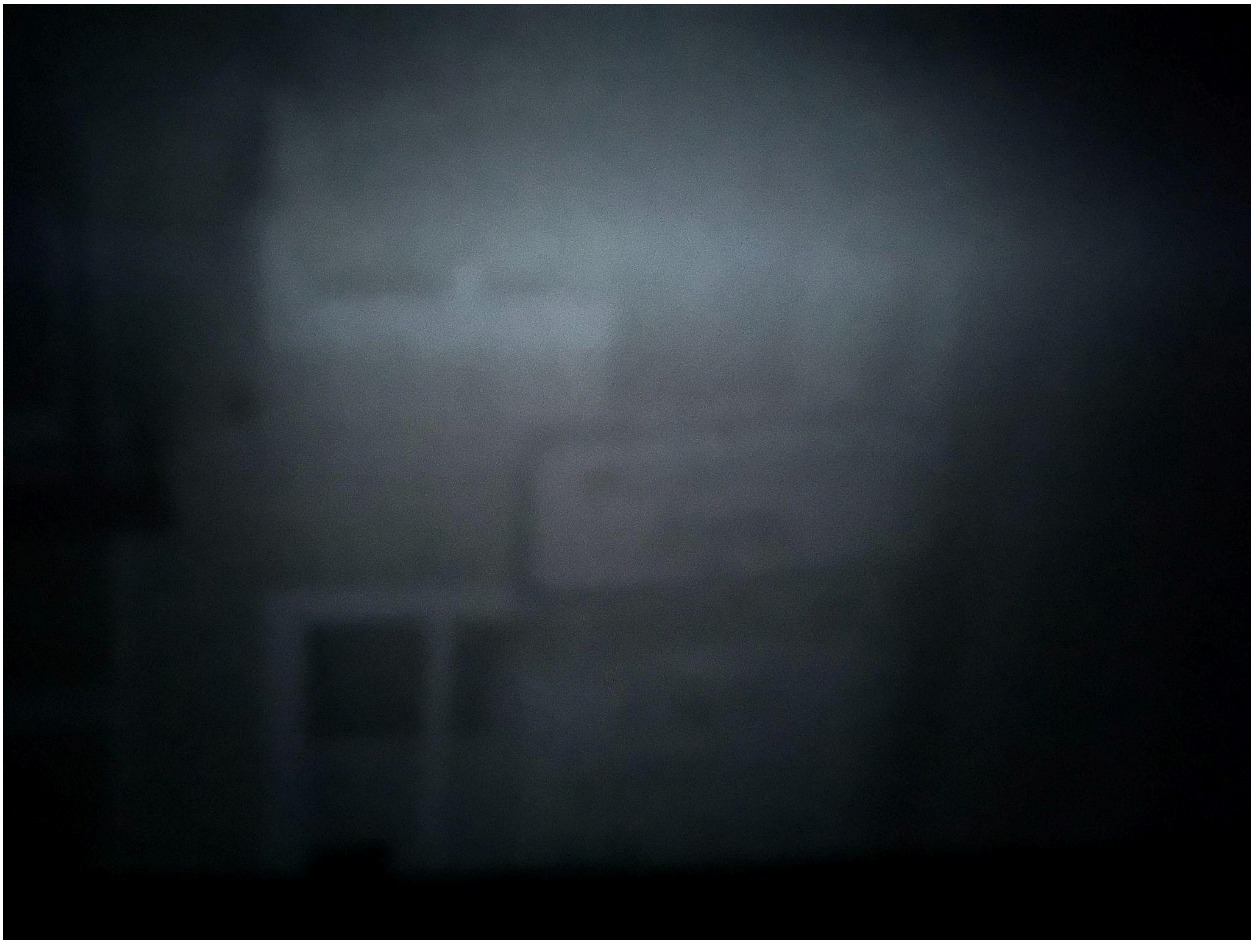


I also made a camera obscura using a dorm room window. This was made by laying thick construction paper completely over the window and poking a large pinhole through to the glass. The image that was captured was crisp but a bit bland, as the window only overlooks some trees:

I used the 0.1mm, 1.0mm, and 5.0mm pinhole sizes for the camera obscura. The images captured with the 0.1mm pinhole size were too dark because not enough light was being passed through to the inside of the box. I captured a parking sign, sky image with a building on the side, and a picture of myself for each set of images:







The 1.0mm pinhole size images appeared to be the best of the three. The images were the most clear and had the best overall picture and brightness:

The 5.0mm pinhole size images came out a bit too blurry and were harder to see than the 1.0mm images. Since enough light got into the box, the images did not come out too dark though: