Assignment 1: RAW Image Processing and Camera Obscura

Andrew id: amangoel

1.1 Image Processing Pipeline

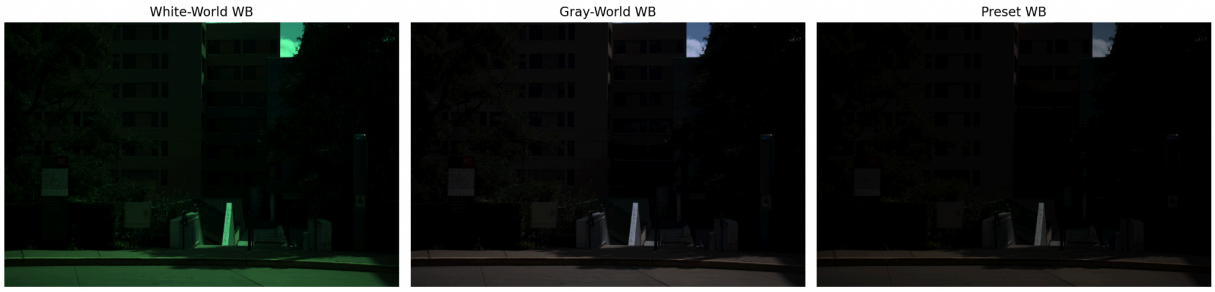
We implemented a complete RAW image processing pipeline in Python, following the instructions in the provided document.

Image parameters:
- Scaling: darkness = 150, saturation = 4095
- Multipliers: 2.394531, 1.000000, 1.597656, 1.000000
- Width: 6016 px
- Height: 4016 px
- Bits per pixel: 16

The pipeline included the following steps: RAW conversion, linearization, demosaicing, white balancing (white world, gray world, and preset methods), color space correction using matrices from dcraw, brightness scaling, and gamma encoding according to the sRGB standard.

For linearization, the black and white levels were set to 150 and 4095, respectively. Demosaicing was performed with scipy's interp2d, supporting the Bayer patterns grbg, rggb, bggr, and gbrg. After testing all four patterns manually, the rggb pattern produced the most consistent colors.

White balancing was tested with three algorithms. The gray world method produced the most neutral colors.



Grayworld and pr setting final images

The two images are in the data as result_grayworld.png and result_pr_setting.png (Need to remove images for pdf

<50mb)

Color correction applied the inverse of MsRGB→cam, constructed as MXYZ→cam multiplied by MsRGB→XYZ. Brightness was scaled to achieve a mean grayscale intensity of 0.25, which appeared most visually pleasing, and gamma encoding followed the sRGB curve with parameters 2.4 and 12.92.

Compression ratio (bytes): 30,945,091 / 7,437,542 = 4.1606610087 for 95 percent quality.

For 75 percent quality compression, the compression ratio was
30,945,091 / 2,542,004 = 12.1735020873. This level of compression looked indistinguishable from the original.

1.2 Manual White Balancing

Manual white balance was implemented using matplotlib.ginput to select two points that defined a white patch. The mean values of the patch were calculated, and gains were applied to normalize the channels so the patch became neutral white. I chose the clouds as the reference for the white balance patch. Refer to manualrectanglewhitebalance.png under data to see which rectangle was chosen.

1.3 Using dcraw

We explored dcraw's flags to replicate our pipeline. The command used:

```
dcraw -a -6 -q 0 -o 1 -H 0 -g 2.4 12.92 campus.NEF
```

Here, -a selects average WB (similar to gray-world), -6 requests 16-bit depth, -q 0 is linear demosaicing, -o 1 converts to sRGB, -H 0 clips highlights, and -g 2.4 12.92 applies the sRGB gamma curve. Conversion from PPM to PNG can be done with pnmtopng.

This is the developed image from draw with these settings. I believe this is the best picture because the clouds are visibly properly whereas gray world image processing pipeline and the campus.jpg provided does not have the cloud distinguishable.

dcraw generated image

The campus.jpg provided image

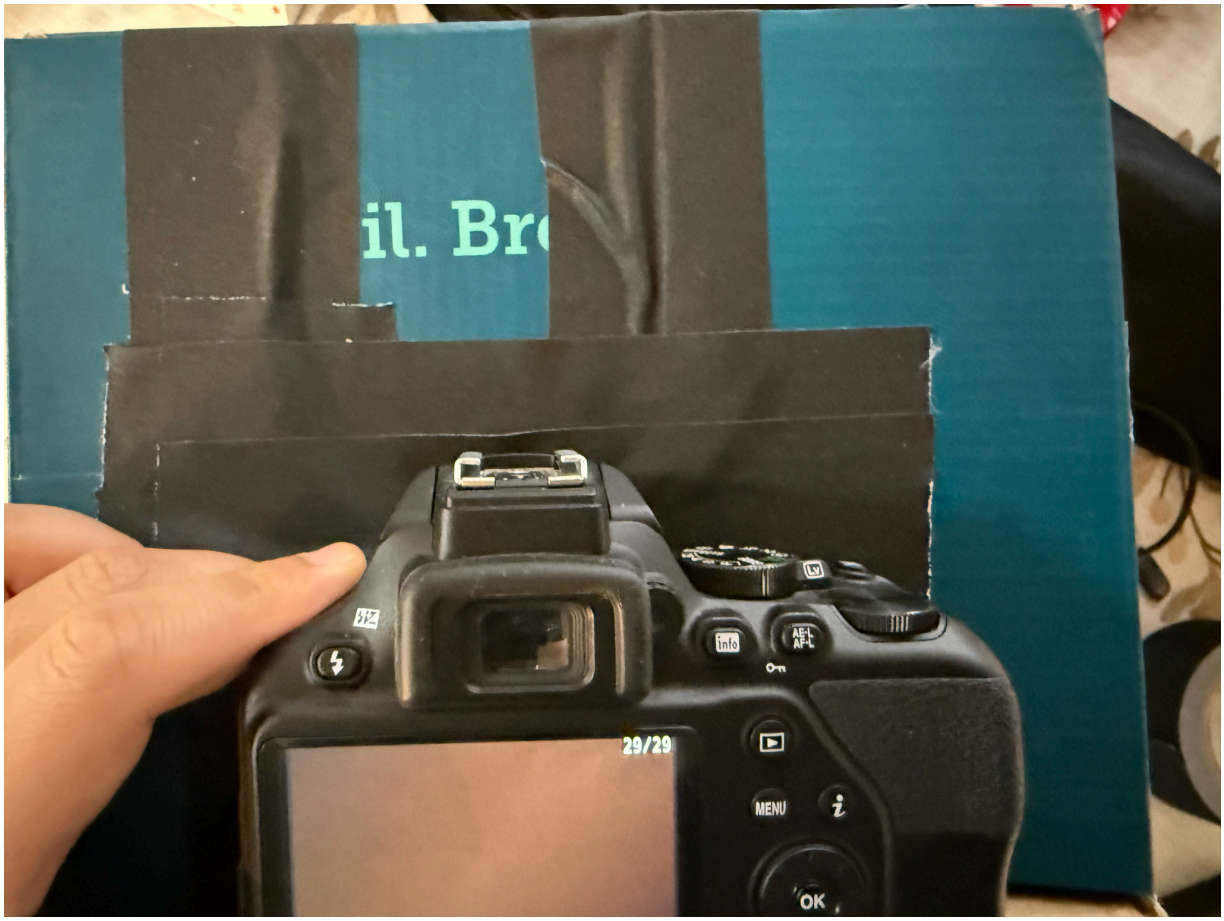
Our result.png

2.1 Camera Obscura

We reviewed instructions for building a pinhole camera. Steps included selecting a shoebox, creating a screen, punching pinholes of different diameters, attaching a DSLR, and capturing images with long exposure. We compared images for different pinhole diameters, noting that smaller holes increase sharpness but require longer exposure. My pinhole camera has a 30 × 30 cm screen and a 50 cm focal length, giving a field of view of about 33°.

LLW·0334
VISITPA.COM


Lysol
AUTONOMY

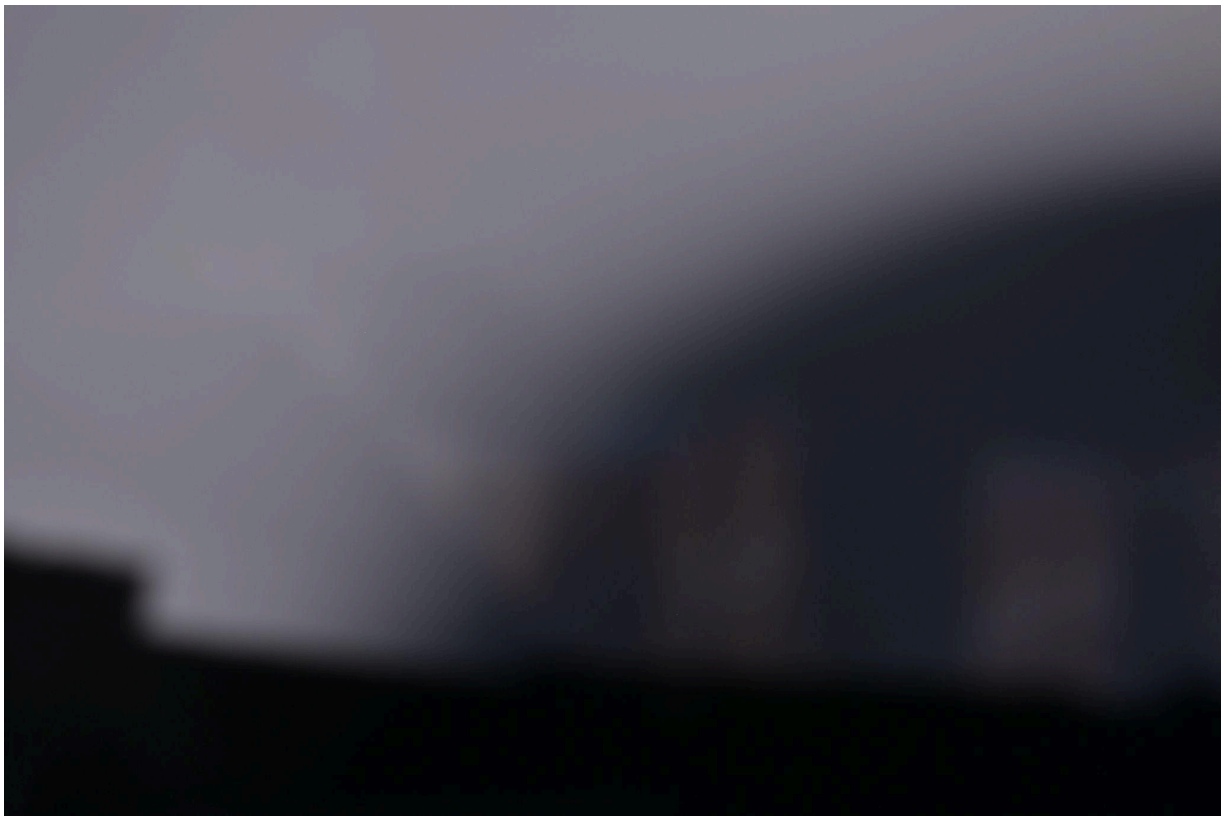


2.2 Camera Obscura

The pinhole size used were 0.1mm, 1mm and 4mm.

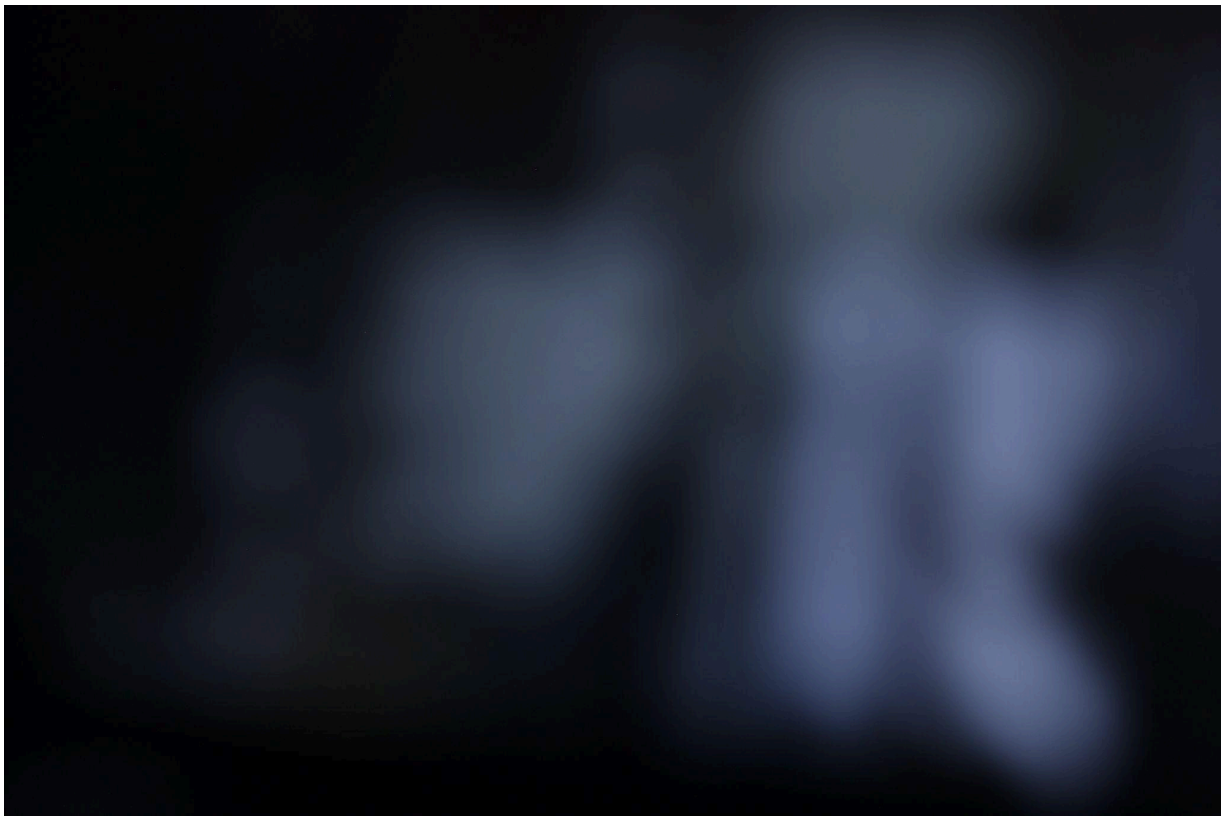
The best picture I could get were of two windows with 1mm pinhole size. The 1mm allowed for sharp details while allowing enough light in during the exposure time.

The second one was with 4mm pinhole.



The 0.1mm pinhole just gave a black screen.

1mm pinhole



4mm pinhole

0.1 mm pinhole

1mm pinhole

4mm pinhole

0.1 mm pinhole

Conclusion

The assignment provided hands-on experience with the full RAW image pipeline and basic optics. The Python pipeline allowed precise control over each step, while dcraw simplified the workflow with a single command. Manual white balance offered creative flexibility, and the pinhole camera demonstrated fundamental imaging principles.