Computational Cameras: Problem Set 1

Bingliang Zhang, UID 2242929, CMS G1

April 16, 2024

Total time spent on this homework: 8h.

1 Building a pinhole camera

The camera setup is depicted in Figure 1. The box camera has dimensions of $16 \times 12 \times 11$ inches, with the 11-inch measurement representing the focal length. To prevent light leakage, black cloth covers the four corners of the screen side. The edges and corners on the pinhole side are also sealed with black paper. The smartphone camera hole is encased in black cloth to avoid light leaks. The white paper on the screen, the black paper on the sides, and the black paper pieces with varying apertures, along with the smartphone, are all securely attached to the box using scotch tape.

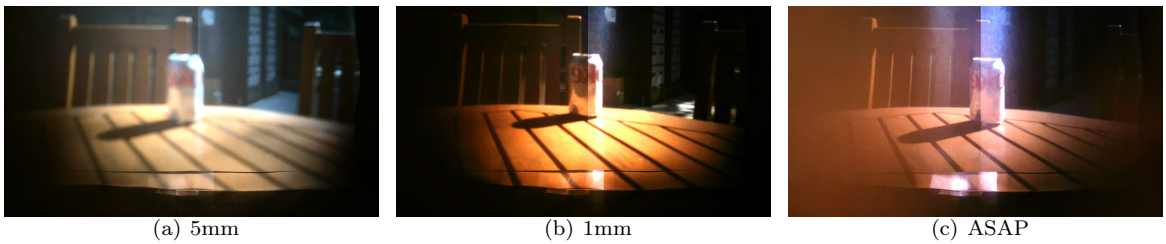
Under the optimal settings for exposure time and ISO, the 1mm aperture delivers the best performance. Photos taken with a 5mm aperture tend to be blurrier, while those from the ASAP aperture appear noisier. This may be due to a certain amount of light leakage inside the box, which results in higher noise levels in the images when aperture is smaller.

(a) (b) (c)

Figure 1: (a) The box pinhole camera. (b) The black paper used to cover edges and corners on the pinhole side. (c) The black cloth used to cover corners on the screen side.

2 Taking pictures

The photos are shown in figure 2, 3 and 4. The exposure time and ISO settings are shown in Table 1. The vertical line running down the middle of the photo marks the boundary between two white A4 papers covering the screen. Some artifacts along the edge of the photo may result from reflections caused by Scotch tape.

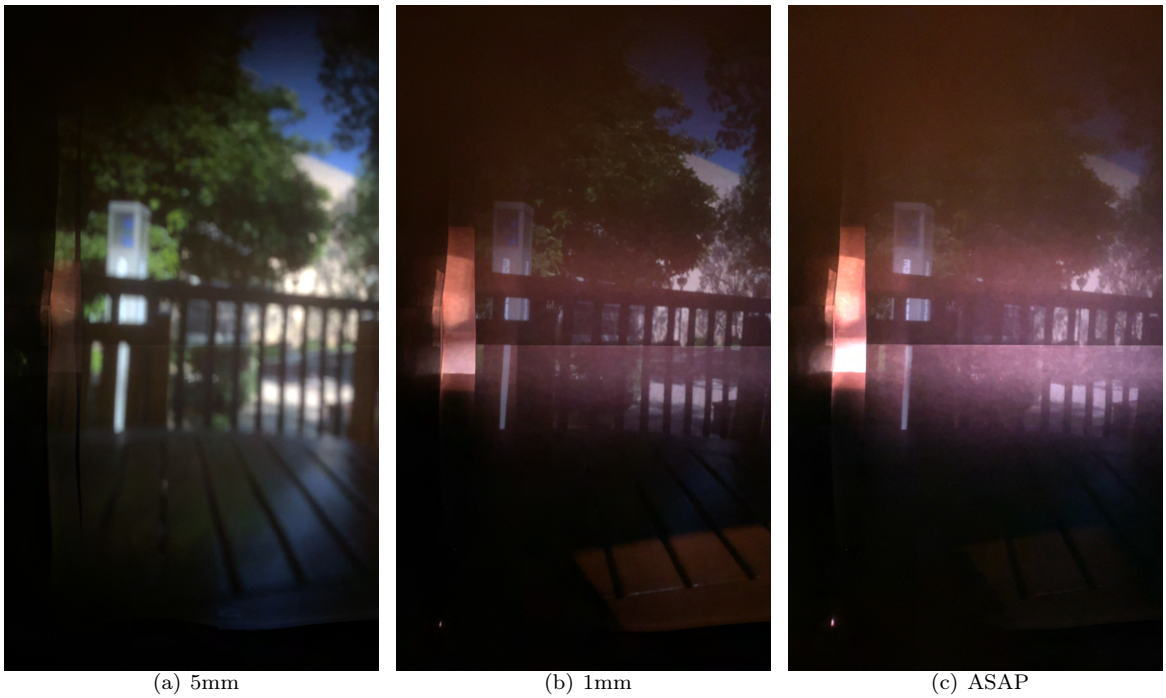


(a) 5mm (b) 1mm (c) ASAP

Figure 2: Photos of the diet coke

(a) 5mm (b) 1mm (c) ASAP

Figure 3: Photos of the parking lot at Cats



(a) 5mm (b) 1mm (c) ASAP

Figure 4: Photos outside the window

Table 1: The hyperparameters in Long Exposure Camera

Aperture	Exposure time	ISO
5mm	6s	400
1mm	10s	400
ASAP	12s	800

3 Image decoding

The code is in function `solve.py/Question3`. The resulting intensity image without gamma encoding is shown in figure 5.

Figure 5: The intensity image

4 Demosaicing and gamma encoding

The code is in function `solve.py/Question4`. The resulting color image (2 pixels smaller) with gamma encoding is shown in figure 6.

Figure 6: The color image

5 The Spanish castle illusion

The code is in function `solve.py/Question5`.

Observations: After staring at a color adaptation image for 20 seconds and then looking at a grayscale image, the grayscale appears to be colored as the original image.

The photos are shown in figure 7.

(a) original

(b) gray scale

(c) adaptation

Figure 7: The photos of Spanish castle illusion.

6 Raspberry Pi setup

The code is in function `solve.py/Question6`.
After adjusting the right focal length of Raspberry Pi Camera, the resulting photo after gamma encoding is shown in figure 8.

Figure 8: A photo from Raspberry Pi Camera

7 Crescents in the shadows

Explanation: The branches of the tree intersect to create numerous gaps, each acting like a small aperture relative to the size of the sun. As sunlight passes through these openings, it projects inverted images onto the ground, resulting in the appearance of crescent-shaped shadows.
The images taken during Monday's solar eclipse in 9.

Figure 9: Crescent shadows of the tree in front of Annenberg.