

Grayscale Camera With CircuitPlayground and a Photoresistor

Group 27: Charlie Schuetz, Uneeb Hyder
Swarthmore College - Computer Engineering Fundamentals

Introduction & Background

Our goal was to create a basic pinhole style camera that could capture grayscale images with reasonable accuracy. A pinhole camera essentially is a light proof box except for one small hole which accepts light. This light then reflects an inverted image on the other side of the box based on the incoming light⁴.

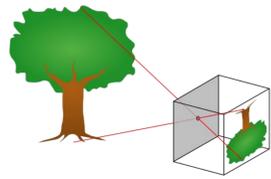


Figure 1. Simple pinhole camera diagram⁹.

Code Example

```
for i in range(51):
    image_row = []
    y=i-30
    for j in range(51):
        if i%2==0:
            x=j+30
        if i%2==1:
            x=80-j
        next_pos=(float(x),float(y))
        alpha, beta = ik(next_pos)
        plotter.set_angles(alpha, beta)
        sleep(TIME_1)

        # Take measurement here
        for k in range(5):
            tests[k]=input.value
            sleep(TIME_2)
```

In this snippet of code, two for loops (one for the vertical direction and the other for the horizontal) collect raw voltage data, with each iteration of the inner loop representing one pixel in the final image, and each iteration of the outer loop representing one horizontal strip of the image. Each iteration of the inner loop moves the arm of the plotter to its next position, takes five measurements, and records the median voltage. In addition, in order to prevent the arm always moving back to the starting level, the horizontal position is altered to make the plotter move in a zigzag pattern.

Design Process

- The pen plotter robot from Lab 3 was used to move a photoresistor.
- The box was sealed with electrical tape, black construction paper, and duct tape to ensure no light from outside could get in (except through the lens).
- A photoresistor circuit with a voltage divider was used to measure the light intensity.
- Our CP code had the pen plotter move in a grid pattern in front of our “lens” and record the light level at each point.
- Using graphics.py, a separate Python file scaled the data and created the image.

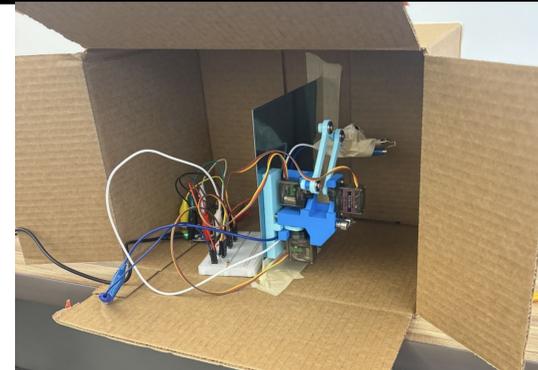


Figure 2. The grayscale camera being built. The pen plotter, breadboard, circuit board, and photoresistor can be seen.

Results

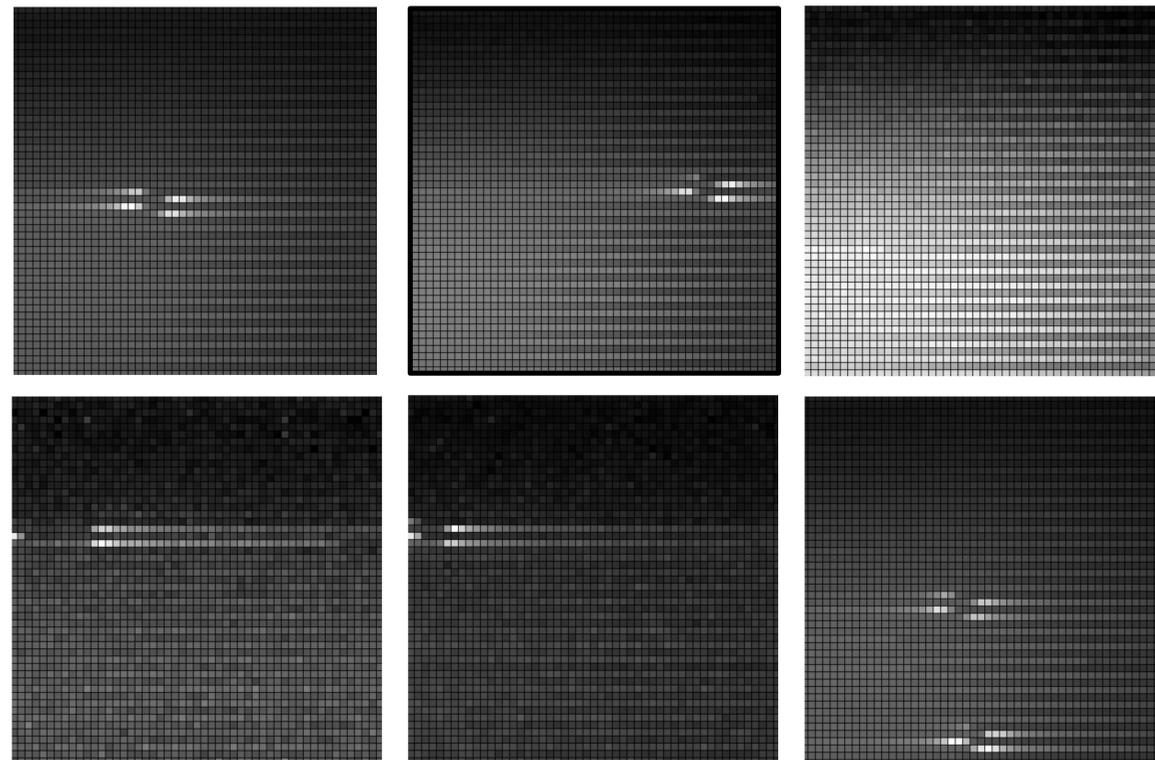


Figure 3. 6 images taken with the camera. The left and middle rows all contain images of a single flashlight in different positions. The right column contains a picture of a large lamp (top) and one of two separate flashlights oriented vertically in line (bottom).

Successes

- The camera is able to detect sources of light with great accuracy.
- However, it is unable to capture object images, likely due to the imprecision of the photoresistor and lens setup.
- The photoresistor appeared to absorb light over time instead of instantaneously, resulting in errors particularly near the bottom of images or a trailing effect immediately after it passed a light source.
- Ultimately, the basic setup is still capable of capturing high contrast images, especially those which contain light sources, implying that further adjustments as described below could improve accuracy and capture object images.

Improvements

- We would try to use a more light proof box with a more accurate pinhole. For example, using a wooden box with a pre-cut pinhole would improve the results.
- A higher quality photocell resistor would also increase accuracy significantly.
- More detailed lens calculations would also improve the results considering the placement of the Circuit Playground setup and the optimal object distance.
- A larger amount of pixels with a larger pen plotter robot would also give substantial improvements.

References

1. Ada, L. (2024, March 8). Using a Photocell. *Adafruit Learning System*. <https://learn.adafruit.com/photocells/using-a-photocell>
2. Ada, L. *Adafruit Learning System*. <https://learn.adafruit.com/assets/459>
3. Ada, L. & Rembor, K. (2025, September 24). Adafruit Circuit Playground Express. *Adafruit Learning System*. <https://learn.adafruit.com/adafruit-circuit-playground-express/pinouts>
4. The Pinhole Camera Model. *Scratch A Pixel*. <https://www.scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera/how-pinhole-camera-work-s-part-1.html>
5. MicroPython & CircuitPython contributors. (2025, December 9). JSON encoding and decoding. *Adafruit CircuitPython Docs*. <https://docs.circuitpython.org/en/latest/docs/library/json.html>
6. Shreyasi, C. (2025, July 23). Reading and Writing lists to a file in Python. *GeeksForGeeks*. <https://www.geeksforgeeks.org/python/reading-and-writing-lists-to-a-file-in-python/>
7. Pieters, M. (2018, September 2). Writing a Large JSON Array To File. *Stack Overflow*. <https://stackoverflow.com/questions/52137608/writing-a-large-json-array-to-file>
8. Daaxix. (2013, January 3). Why doesn't my pinhole camera work? *Physics Stack Exchange*. <https://physics.stackexchange.com/questions/48243/why-doesnt-my-pinhole-camera-work>
9. Wikipedia Contributors. (2025, April 17). Pinhole camera model. *Wikipedia*. https://en.wikipedia.org/wiki/Pinhole_camera_model