

# CS 484: Introduction to Computer Vision

*Spring 2024*

## Homework 2

### Modules/Libraries Used

**Pathlib** and **os** modules are used for loading images. **Numpy** is used for mathematical operations. **Matplotlib** is used for reading, displaying images and plotting. **OpenCV** is used for color conversions, normalization of pixel values, Canny edge detection, probabilistic Hough transform, and drawing on the images.

### The Data

Template Images



Figure 1: Sample template images.

Grayscale Template Images



Figure 2: Sample grayscale template images.

Rotated Images

**Figure 3:** Sample rotated images.

Grayscale Rotated Images

**Figure 4:** Sample grayscale rotated images.

The data contains 15 template and 15 rotated book cover images. Images in the rotated set are rotated counterclockwise by between 0 and 180 degrees.

## Algorithm Parameter Values

Two sets of parameter values will be investigated in this report.

	"Canny Parameter"	Hough Threshold	Number of Bins
Set 1	100	50	8
Set 2	30	40	12

**Table 1:** Parameter value sets to be investigated.

The canny edge detection algorithm requires upper and lower threshold values. The "Canny Parameter" in Table 1 sets the lower and upper threshold values according to the following rule:<sup>1</sup>

Lower threshold = "Canny Parameter"

Upper threshold = 3 \* "Canny Parameter"

Other parameters are self explanatory.

<sup>1</sup> <https://stackoverflow.com/a/25125926>

# Canny Edge Detection

Canny Edge Detected Template Images



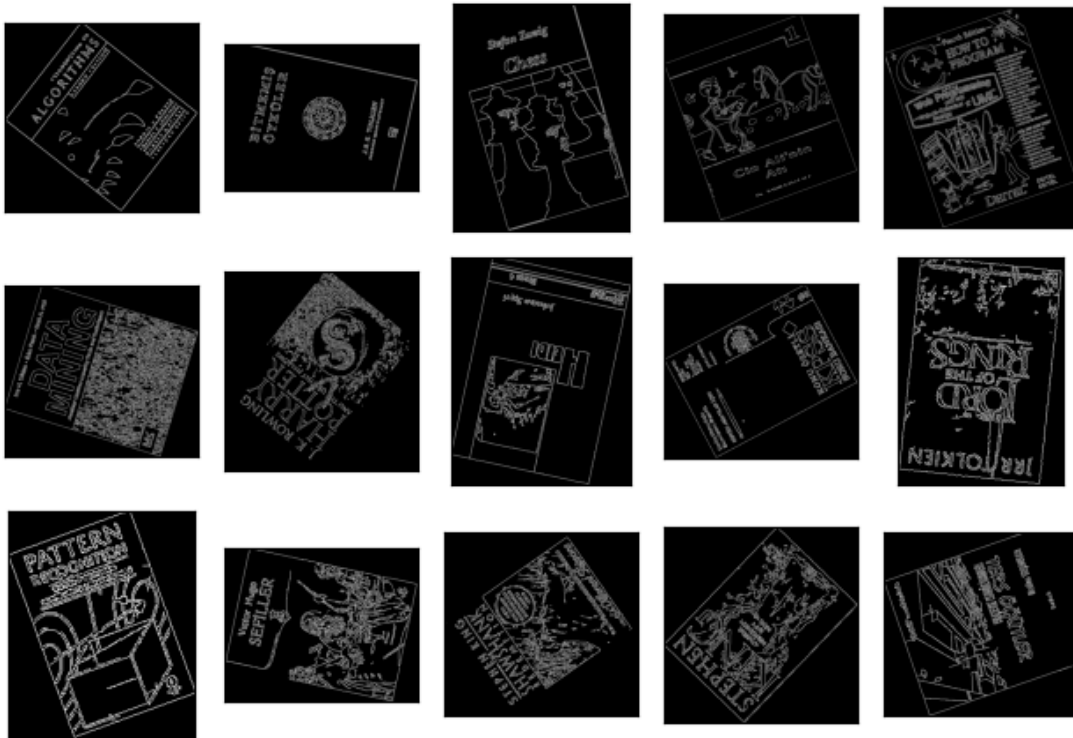
Figure 5: Canny edge detection results of all template images using parameter set 1.

Canny Edge Detected Template Images



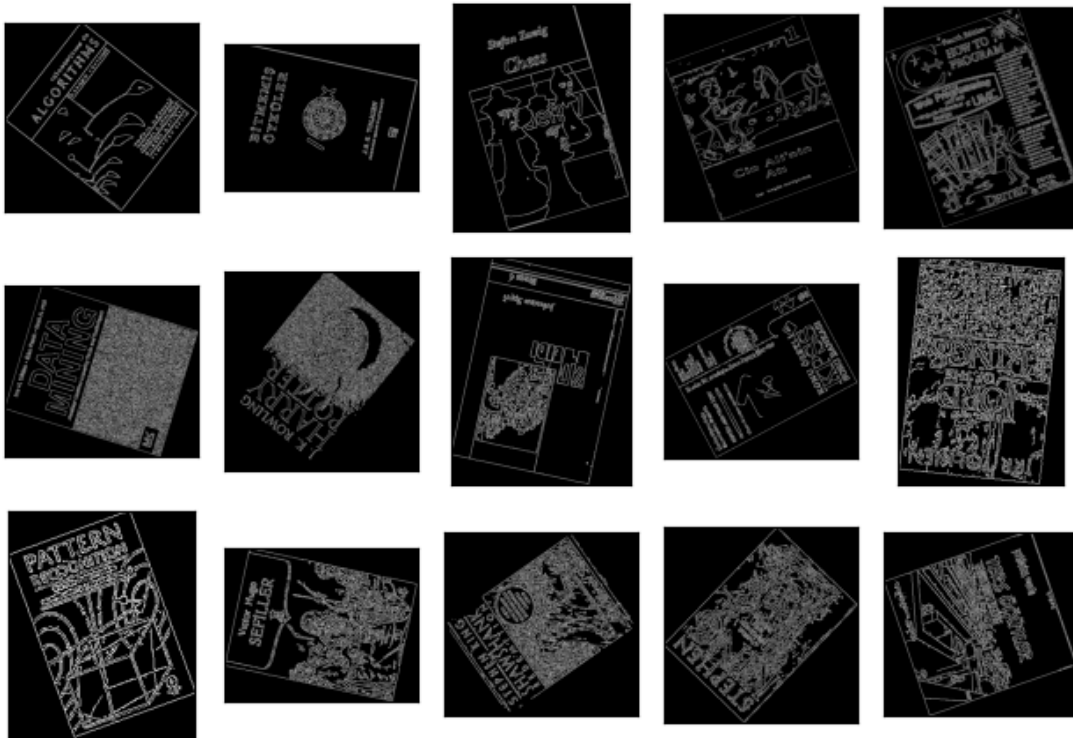
Figure 6: Canny edge detection results of all template images using parameter set 2.

### Canny Edge Detected Rotated Images



**Figure 7:** Canny edge detection results of all rotated images using parameter set 1.

### Canny Edge Detected Rotated Images



**Figure 8:** Canny edge detection results of all rotated images using parameter set 2.

Figures 5 and 6 are the results of running the Canny edge detection algorithm on the grayscale template images using parameter set 1 and 2. Similarly, figures 7 and 8 are the results of running

the Canny edge detection algorithm on the grayscale rotated images using parameter set 1 and 2. **Parameter set 1 produces images with fewer edges compared to parameter set 2.**

## Line Fitting with Probabilistic Hough Transform

Template Images with Lines



**Figure 9:** Line fitting results of 4 sample template images using parameter set 1.

Template Images with Lines



**Figure 10:** Line fitting results of 4 sample template images using parameter set 2.

Rotated Images with Lines



**Figure 11:** Line fitting results of 4 sample rotated images using parameter set 1.

Rotated Images with Lines

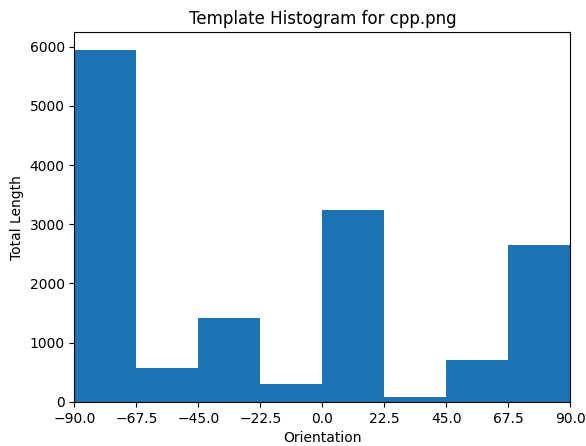


**Figure 12:** Line fitting results of 4 sample rotated images using parameter set 2.

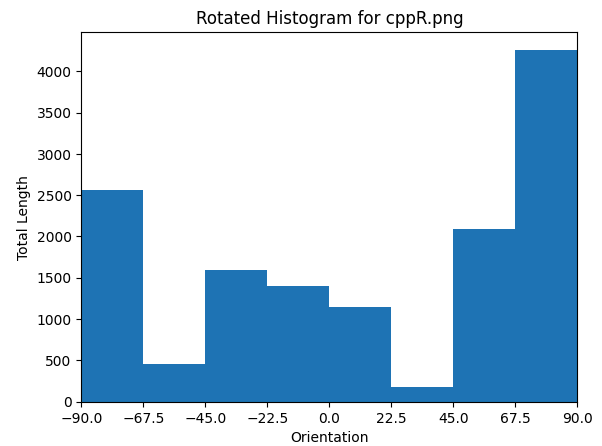
Figures 9 and 10 are results of running the probabilistic Hough transform on the Canny edge detected template images using parameter set 1 and 2. Similarly, Figures 11 and 12 are results of running the probabilistic Hough transform on the Canny edge detected rotated images using parameter set 1 and 2. **Parameter set 1 produces fewer detected edges compared to parameter set 2.**

# Line Orientation Histograms

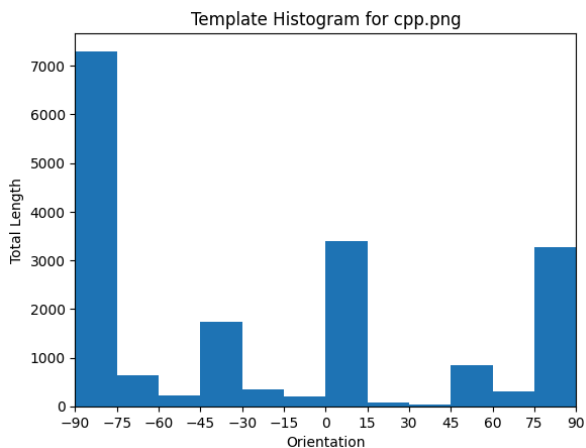
In this section, the image of the cover of the book “C++ How to Program” will be used as an example. All the histograms are available in the .ipynb file.



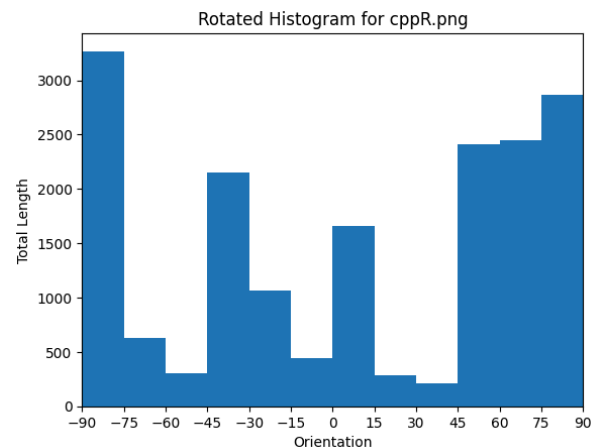
**Figure 13:** Sample template image line orientation histogram weighted by line length obtained with parameter set 1.



**Figure 14:** Sample rotated image line orientation histogram weighted by line length obtained with parameter set 1.



**Figure 15:** Sample template image line orientation histogram weighted by line length obtained with parameter set 2.



**Figure 16:** Sample rotated image line orientation histogram weighted by line length obtained with parameter set 2.

Figures 13 and 14 are the weighted line orientation histograms of template and rotated images of the book using the parameter set 1. Similarly, figures 15 and 16 are the weighted line orientation histograms of template and rotated images of the book using the parameter set 2. **Parameter set 2 has more histogram bins compared to parameter set 1. This results in more precise rotation angle estimation. Additionally, the weighted line orientation histograms of rotated images do not exactly resemble circular shifted versions of weighted line orientation histograms of template images. This can be due to the limited resolution of the rotated images.**

## Template to Rotated Image Matching

For each template image, first, circular shifts are used to select the histogram with the lowest euclidean distance for each rotated image. Then, the selected histograms of rotated images are



compared and the rotated image that has the histogram with the lowest euclidean distance to the histogram of the template image is selected.

The results are as follows for **parameter set 1**:

- The best match for `algorithms.png` is `algorithmsR.png` with an angle of `66` degrees counterclockwise rotation.
- The best match for `bitmemisoykuler.png` is `kpssR.png` with an angle of `44` degrees counterclockwise rotation.
- The best match for `chess.png` is `kpssR.png` with an angle of `132` degrees counterclockwise rotation.
- The best match for `cinali.png` is `cinaliR.png` with an angle of `22` degrees counterclockwise rotation.
- The best match for `cpp.png` is `cppR.png` with an angle of `22` degrees counterclockwise rotation.
- The best match for `datamining.png` is `dataminingR.png` with an angle of `88` degrees counterclockwise rotation.
- The best match for `harrypotter.png` is `harrypotterR.png` with an angle of `132` degrees counterclockwise rotation.
- The best match for `heidi.png` is `kpssR.png` with an angle of `132` degrees counterclockwise rotation.
- The best match for `kpss.png` is `kpssR.png` with an angle of `132` degrees counterclockwise rotation.
- The best match for `lordofrings.png` is `chessR.png` with an angle of `22` degrees counterclockwise rotation.
- The best match for `patternrecognition.png` is `chessR.png` with an angle of `110` degrees counterclockwise rotation.
- The best match for `sefiller.png` is `sefillerR.png` with an angle of `88` degrees counterclockwise rotation.
- The best match for `shawshank.png` is `shawshankR.png` with an angle of `44` degrees counterclockwise rotation.
- The best match for `stephenking.png` is `stephenkingR.png` with an angle of `132` degrees counterclockwise rotation.
- The best match for `ters.png` is `lordofringsR.png` with an angle of `88` degrees counterclockwise rotation.

**Score:** 9 / 15

The results are as follows for **parameter set 2**:

- The best match for `algorithms.png` is `algorithmsR.png` with an angle of `60` degrees counterclockwise rotation.
- The best match for `bitmemisoykuler.png` is `bitmemisoykulerR.png` with an angle of `90` degrees counterclockwise rotation.
- The best match for `chess.png` is `algorithmsR.png` with an angle of `150` degrees counterclockwise rotation.
- The best match for `cinali.png` is `lordofringsR.png` with an angle of `90` degrees counterclockwise rotation.
- The best match for `cpp.png` is `stephenkingR.png` with an angle of `135` degrees counterclockwise rotation.

- The best match for `datamining.png` is `dataminingR.png` with an angle of `90` degrees counterclockwise rotation.
- The best match for `harrypotter.png` is `harrypotterR.png` with an angle of `135` degrees counterclockwise rotation.
- The best match for `heidi.png` is `sefillerR.png` with an angle of 90 degrees counterclockwise rotation.
- The best match for `kpss.png` is `kpssR.png` with an angle of `120` degrees counterclockwise rotation.
- The best match for `lordofrings.png` is `sefillerR.png` with an angle of 90 degrees counterclockwise rotation.
- The best match for `patternrecognition.png` is `algorithmsR.png` with an angle of 60 degrees counterclockwise rotation.
- The best match for `sefiller.png` is `kpssR.png` with an angle of 120 degrees counterclockwise rotation.
- The best match for `shawshank.png` is `shawshankR.png` with an angle of `45` degrees counterclockwise rotation.
- The best match for `stephenking.png` is `stephenkingR.png` with an angle of `135` degrees counterclockwise rotation.
- The best match for `ters.png` is `lordofringsR.png` with an angle of 90 degrees counterclockwise rotation.

**Score:** 7 / 15

**Parameter set 1 scores better than parameter set 2 with 9/15 vs 7/15 correct matches. Both parameter sets guess “The Shawshank Redemption” correctly but report the angle incorrectly.**

## Conclusion

The method implemented in this homework is extremely sensitive to parameters. Summarizing parameter effects, the “Canny Parameter” is **inversely correlated with the number of detected edges**. Similarly, the threshold parameter of the Hough transformation is **inversely correlated with the number of detected lines**. While a larger number of bins increase rotation angle resolution, this parameter must be chosen to harmonize with the other two parameters. The effectiveness of the parameters heavily depend on each other. While one value may work well for one set of parameters, the same value might not work as well for other values of the other parameters.