CMP_SC 7650: Digital Image Processing
Homework 6: Review of Image Processing Operations
By: Mikey Joyce
Due: 11/21/2023

Abstract:

This assignment is designed to let us experiment with different geometric transforms and perform edge detection in a controlled environment. Only one test image was given, it was a picture of Naka Hall.

Introduction:

In this assignment, given the test image, various geometric transformations were performed (translate, cropscale, vertical flip, horizontal flip, rotate, and fill). I opted to do these transforms by hand rather than using an outside library to do the transforms. After the geometric transforms are produced, the next experiment will be performing edge detection on the building.

Experiments and Results:

Original Image

Experiment A:

Translate

Crop Scale

Vertical Flip

Horizontal Flip

Rotate

Fill

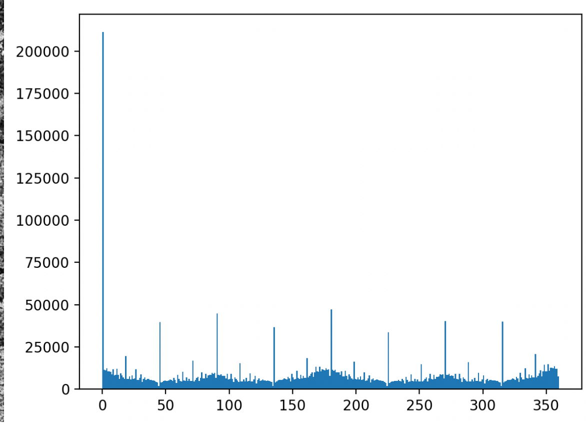
Experiment B:

Small Scale:

Ix

Iy

M

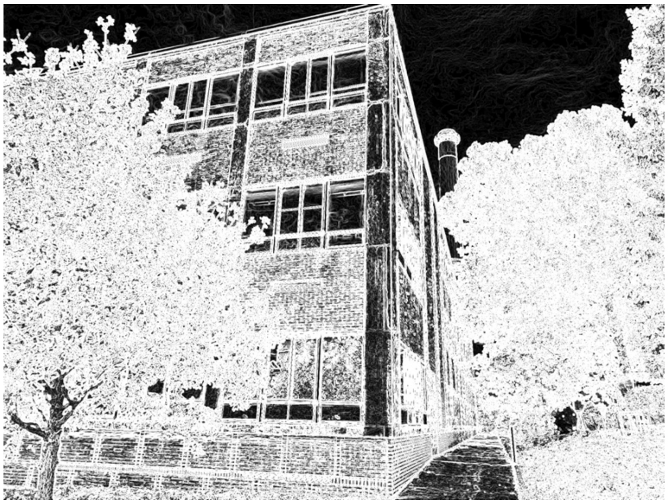


Histogram of angles

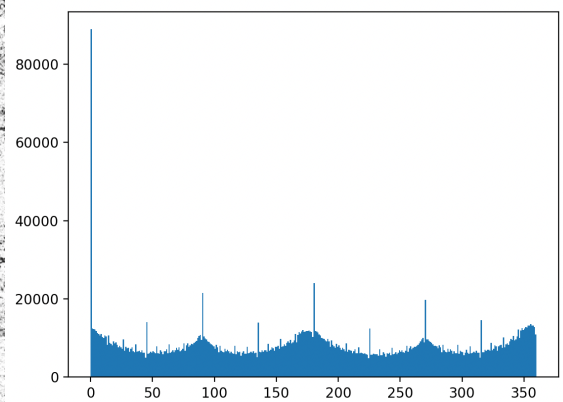
Large scale:

Ix

Iy



M



Histogram of angles

Conclusion:

In this homework assignment I successfully implemented the following geometric transforms from scratch: translate, cropscale, vertical flip, horizontal flip, rotate, and fill. Since it was done without using libraries other than numpy it is possible that the computations take some time (no longer than 30s for any given transform). It was fun to see how to transforms effected the image. Part B of the homework assignment revolved around edge detection. To detect the edges, I picked two scales. I interpreted "scale" as the dimensions of the sobel kernel. So, the small scale I picked a sobel kernel of size 3x3 and for the large scale I picked a sobel kernel of size 5x5. Using these sobel kernels I convolved them over the image to get the first derivative in the horizontal direction (Ix) and the vertical direction (Iy). Using Ix and Iy, I then computed the magnitude M using the edge gradient equation provided in Lec08_EdgeDetection.pdf. Lastly, I implemented the angle equation on the same slide. At first, I used np.arctan() but realized to get the correct computation math.atan2 needed to be used as was mentioned in the homework instructions. For the small scale it was able to detect the fine details in the image, I think the resulting image M looks quite crisp. For the larger scale, it didn't seem to detect large structures, but seemed to just make the detection brighter in contrast. Maybe using a much larger dimension size for the sobel kernel would give a better detection of large structures. It was also nice to note how the histograms changed, the large scale histogram looks more smooth.

References:

- Libraries and tools: PyCharm, OpenCV, NumPy, math, scipy, Matplotlib, Preview
- Lec08_EdgeDetection.pdf
- Lec11_GeometricTransforms.pdf