CSC 340: Programming Assignment 3 – Harris Corners
Hannah Moore

1. **Assignment Description**

The goal of this assignment is to write a program that extracts corner features from a photograph. The program uses Harris Corners algorithm along with 2 different ranking corners techniques.

2. **Harris Corners Algorithm**

The Harris Corners Algorithm is used for detecting corners in any given image. The algorithm looks at each pixel in the image and calculates the Ixx, Iyy and Ixy values. Next, all the Ixx, Iyy and Ixy values are each summed up and formed into a matrix. The determinant and trace of the matrix are then computed and used in a formula to calculate the c values, or the measurement of corner response. After the c values are computed, there are 2 corner ranking techniques that can be used to determine which c values are corners.

The first method that can be used looks at all the c values and finds the largest value. Next, all of the c values are looked at again to find values that are greater than a certain percentage of the largest c value. Those c values are considered corners.

The second method first divides the image into certain size sections. Next, each c value within each section is looked at to determine the top N largest c values within each section. Those c values are considered the corners.

CSC 340: Programming Assignment 3 – Harris Corners
Hannah Moore

3. Equations Used

a. Calculating the Ixx, Iyy and Ixy values

- Ix(a, b) = image(a+1, b) – image(a-1, b)
- Iy(a, b) = image(a, b+1) – image(a, b-1)
- Ixx(a, b) = (Ix)^2
- Iyy(a, b) = (Iy)^2
- Ixy(a, b) = Ix * Iy

b. Computing the determinant of the matrix

- $(M_{11} * M_{22}) - (M_{12} * M_{21})$

c. Computing the trace of the matrix

- $M_{11} + M_{22}$ (Sum of the diagonals)

d. Calculating the c values

- C = determinant – 0.05 * (trace^2)

CSC 340: Programming Assignment 3 – Harris Corners
Hannah Moore

4. Images

Figure 1.1

- Method 2
- Parameters: *M* blocks = 15x15, *N* pixels = 3

Figure 1.2

- Method 1
- Parameter: 5%

CSC 340: Programming Assignment 3 – Harris Corners
Hannah Moore

Figure 2

- Method 1
- Parameter: 15%

CSC 340: Programming Assignment 3 – Harris Corners
Hannah Moore

Figure 3

- Method 1
- Parameter: 2%

CSC 340: Programming Assignment 3 – Harris Corners
Hannah Moore

Figure 4

- Method 2
- Parameters: *M* blocks = 4x4, *N* pixels = 200

Figure 5

- Method 2
- Parameters: *M* blocks = 2x10, *N* pixels = 40

CSC 340: Programming Assignment 3 – Harris Corners
Hannah Moore

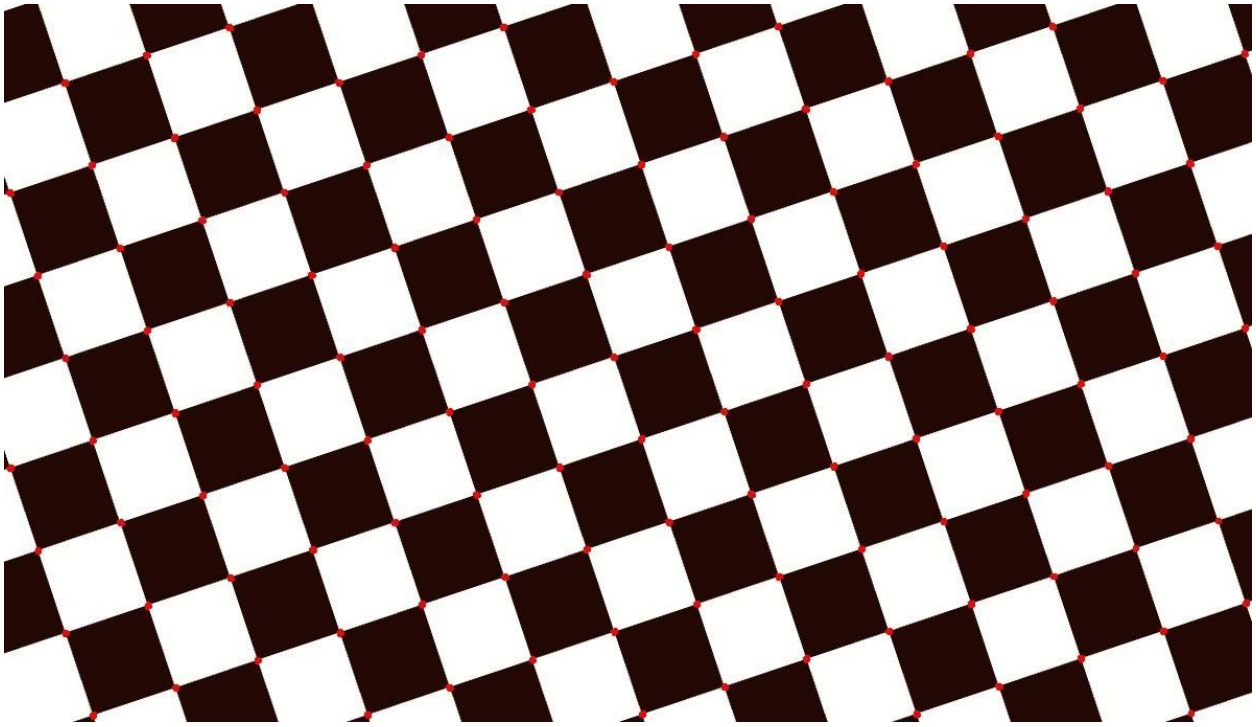

Figure 6

- Method 1
- Parameter: 20%

Observations:

While implementing Harris Corners using 6 different images, I notice that one of the methods would always work better than the other depending on the image. Method 1 seemed to find the corners better on images that were simple. In other words, images that did not contain a lot of variation in detail and that had obvious corners. Method 2 appeared to work better on the opposite kind of images, ones with a lot of detail and not so obvious corners. I believe this is because method 2 allows for more control since you have 3 parameters to change rather than 1.

5. Issues Encountered

While working with method 2 on various images, I found it difficult to find all the corners perfectly. It took some time messing around with different combinations of parameters to get a majority of the corners. Also, while implementing this method, it took the longest amount of time to figure out how to loop through the image by sections and then create a loop that looks through each section. Lastly, I did not realize right away that when referencing the image of c values, created from computing the values using the formula determinant – k(trace)^2, and the Ixx, Iyy and Ixy images, you must add a third index value of 0 to get the correct value.