Mathew Seedhom
CS 558
Homework 1
I pledge my honor that I have abided by the Stevens Honor System.

1. Pre-Processing

   In order to pre-process the image I first applied the Gaussian filter to blur out noise. I did this by defining a Gaussian Kernel, and creating a function to apply any filter that is given to an image. I then applied Sobel Kernels to the blurred image with the same function, and applied thresholding and non-maximum suppression on it in order to have the image fit in the background of my future images. To apply the Hessian Matrix, I applied each Sobel filter on the pictures that were already convolved with the Sobel Kernels to get derivative approximations for the image at each pixel. After using the Hessian Matrix, I thresholded the pixels in the resulting image in order to only keep the 10% of pixels with the highest intensities. I then applied non-maximum suppression to this to keep pixels with the highest intensities within a 3x3 area around it. I then connected this with my edge image, in order to show the pixels with some relation to the original image.

2. RANSAC

   I applied the RANSAC algorithm to my image with the corners in order to find relations between points. I would randomly choose 2 points from a specified set of pixels, and create a line around it. I then checked which corners fit in a specified distance within the line by using geometric properties of linear functions. More specifically, I found the equation of the line between the two random points, and made a new linear equation to represent the shortest distance between each corner to the line. If the calculated distance fell within the range I stored it and continued. I applied this a specified amount of times and took the line with the most points near it. I then removed those points and repeated this 3 more times. To draw the lines I took the points that were nearest the line and computed their distance from the two originally computed edge points. I then drew a line between the points that had the farthest distance from one of the two edge points while their distance to the other edge point was greater than the distance between the original edge points. This was used to maintain some form of orientation to the line.

3. Hough Transform

   For the Hough Transform I set the radius of the image and the angle to the x-axis into discrete bins, and surveyed each pixel for these bins to find if there exists a line that would cross that point. I repeated this for each corner, while storing the bins with the most votes for future use. Once I had all lines that would pass through the most points, I drew lines that passed through the y axis, and the line that intersects the x axis at the edge of the image.