Jeremy Midvidy, jam658
EECS 332, Fall 2018
MP5 Report

About My Code:

My code for MP5 can be found in this folder, in the python file, `MP5.py`. The file is callable from the command line with the command `$ MP5.py PATH_TO_IMAGE`.

After running, it will write many result images (after each step) to the folder, "results."

I wrote my code in Python 3.7, and it assumes python packages `numpy`, `sys`, `matplotlib`, `os`, `math`, `skimage`, `copy`, `scipy`, `and sys` are installed.

Results:

I ran my code on the given image files, and in short, my code performed pretty well. The code reads the image, and then executes this logic:

(1) Gaussian Smoothing
- (a) Applies standard gaussian smoothing with a given sigma to the inputted image. This process removes a lot of edges and sharpness from the given image.

(2) Calculate Image Gradient
- (a) Using the smoothed image, I use a Robert Cross filter to calculate the image gradient. First I make two gaussians, Gx and Gy, from the convolution of the given image and either the vertical rc mask or the horizontal rc mask, respectively. Then I resolve the final gaussian, and return the theta and magnitude values such that:

$$\theta = \sqrt{G_x{}^2 + G_y{}^2}$$

$$mag = arc\tan 2(Gy, Gx)$$

(3) Select High and Low Thresholds
- (a) I gather T_Low and T_High from the magnitude (gradient) array using a quantized histogram.

(4) Suppress Non Maxima
- (a) I iterate through the image. For each cell, I gather the line that corresponds to its angle in the gradient (specifically: `line =int(theta[r, c] % 4`). This line value corresponds to: {0: Horizontal, 1: Upleft-Downright Diagonal, 2: Vertical, 3: Upright-Downleft Diagonal}.
- (b) Then I check to see if the given cell is the local maxima through each of its neighbors along the given line. If it is, I keep its value. If it is not, I change its value to 0.

(5) Thresholding and Edge Linking
- (a) I iterate through the image. For each cell, I see whether it is greater than T_High. If it is, then I set it's color value to 255. Then if `flag="TLOW"`, I check whether any of its 8 neighbors are above the T_Low threshold. If it is, I set the neighbors value to 255.
- (b) If `flag="THIGH"`, then I do not check the to see if the neighbors are above T_Low.

I experiment with different input values, parameters, input images, and edge detectors to compare outcomes. Results can be seen below.

Results Analysis for Lena.bmp

Input	Gaussian Smoothing	Theta



Gradient	Robert Cross	Suppressing Nonmaxima



T_Low	T_High	Parameters


T_Low = 16.3973
T_High = 32.7947
Sigma = .5
Filter Used: Robert Cross

Results Analysis for gun1.bmp

Input	Gaussian Smoothing	Theta



Gradient	Robert Cross	Suppressing Nonmaxima



T_Low	T_High	Parameters


T_Low = 8.4624
T_High = 16.9248
Sigma = .5
Filter Used: Robert Cross

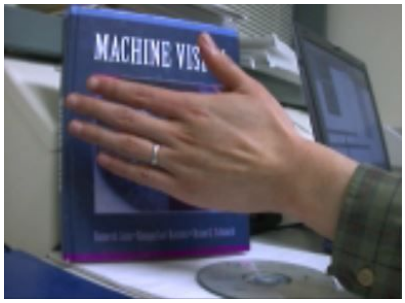
Results Analysis for joy1.bmp

Input	Gaussian Smoothing	Theta
		

Gradient	Robert Cross	Suppressing Nonmaxima
		

T_Low	T_High	Parameters
		T_Low = 21.8532 T_High = 47.7064 Sigma = .5 Filter Used: Robert Cross

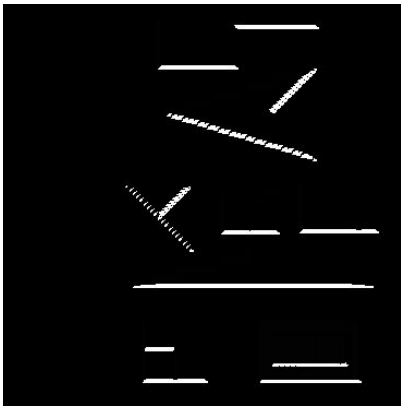
Results Analysis for pointer1.bmp

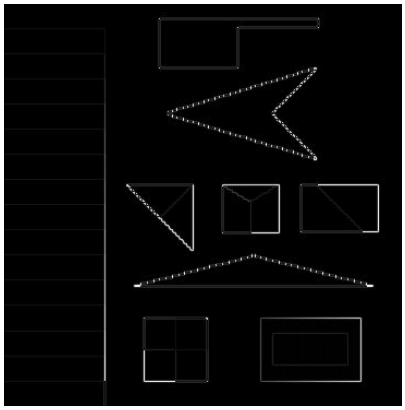
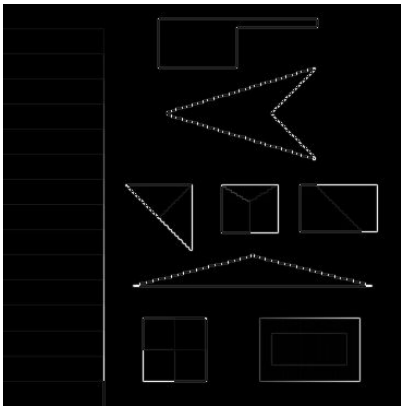
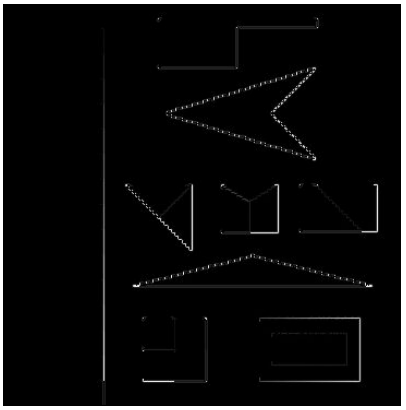
Input	Gaussian Smoothing	Theta
		

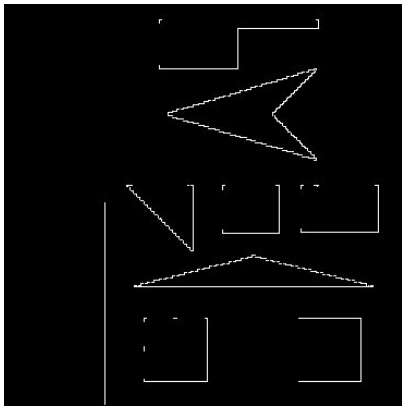
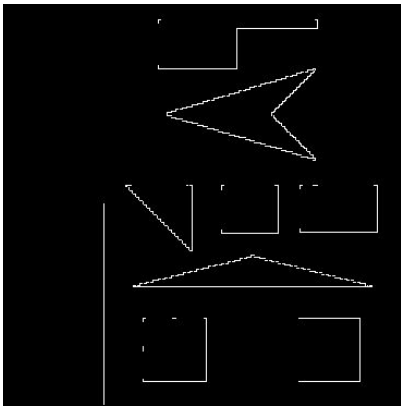
Gradient	Robert Cross	Suppressing Nonmaxima
		

T_Low	T_High	Parameters
		T_Low = 20.0995 T_High = 41.1990 Sigma = .5 Filter Used: Robert Cross

Results Analysis for test1.bmp

Input	Gaussian Smoothing	Theta
		

Gradient	Robert Cross	Suppressing Nonmaxima
		

T_Low	T_High	Parameters
		T_Low = 62.7036 T_High = 125.4072 Sigma = .5 Filter Used: Robert Cross