Jeremy Midvidy, jam658
EECS 332, Fall 2018
MP6 Report

About My Code:

My code for MP6 can be found in this folder, in the python file, `MP6.py`. The file is callable from the command line with the command `$ MP6.py PATH_TO_IMAGE`.

After running, it will write a result image named "`PATH_TO_IMAGE_lines`" of the line detected image to the folder named "results" as well as print the greyscale, gaussian, canny edge, and hough-parameter space. Additionally, it will print relevant [theta, p] values to the command line.

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

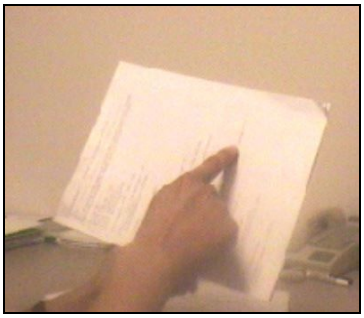
Results:

In short, my code performed well on the given test cases. My code reads the image into an np array of pixel values and then executes this logic:

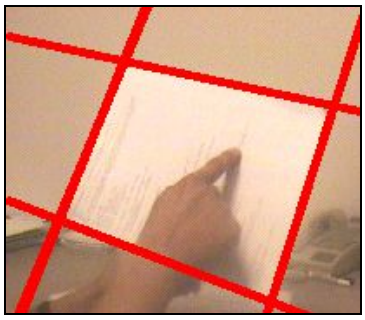
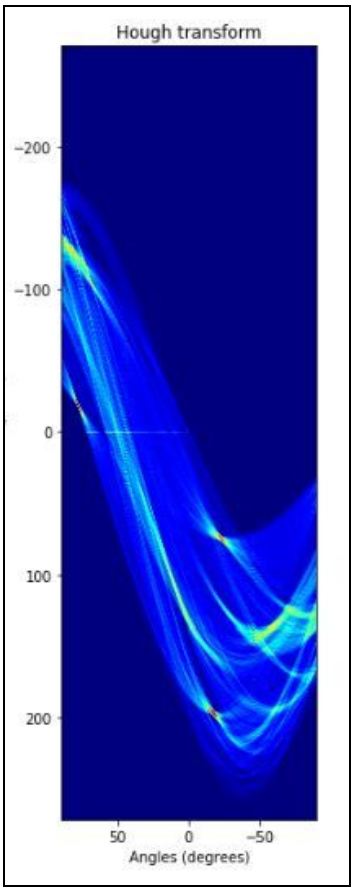
1) Apply a gaussian filter to smooth the image.
2) Get the edges of the image using my Canny Edge detector from MP5.
3) Get the Hough-Transform by:
    a) Initializing the parameter space to be a 2-D matrix of bins where rows represent theta values from [0, 180) (degrees) and columns represent rho values from [0, 2*Dmax+2), where Dmax = $\sqrt{R^2 + C^2}$ where R and C are the number of rows and cols in the input image, respectively.
    b) Then, for each edge detected by the Canny Edge detector (every non 255 cell in the edge image), I "vote" for a given [theta, p] by calculating the p for every possible theta value between [-90, 90), where $\rho = xcos(\theta) + ysin(\theta)$ and (x,y) are coordinates of the current cell.
        i) For every vote, I zero-index it in the 2-D parameter space matrix by adding 90 to it's theta value and Dmax to its rho value. This transformation is useful for maintaining the parameter matrix. Later, I undo this transformation when finding lines.
    c) Then, after the parameter space matrix votes are cast, I find the local maxima. To do this, for every bin in the parameter 2-D matrix, I create a 60x60 mask (mask size can be tuned with the `threshold` variable). If the current cell is the max value in the mask, then it is a local maxima. If it is a local maxima, I append it's [theta, p] value to a list and continue. If it is not a local maxima, I continue.
        i) With the local maxima list constructed, I pass that list as well as a few parameters to the final `drawLines()` function.
    d) Finally, to draw the detected lines, I iterate through the list of local maxima [theta, p] tuple values. For each tuple, make a translation from polar [theta, p] to cartesian [x,y] such that I know have an equation of a line y=mx+b (this can be achieved by rearranging the equation for rho in (b) such that y is isolated on the left side.
        i) With y=mx+b, I create two y values for x1=1, x2=1000, and then plot the line between (x1,y1), (x2, y2) on the graph. These lines represent my detected lines from the Hough-Transform.

Results Analysis for Input.bmp

Input	Greyscale	Gaussian



Canny Edge	Line Detection	Hough Parameter Space

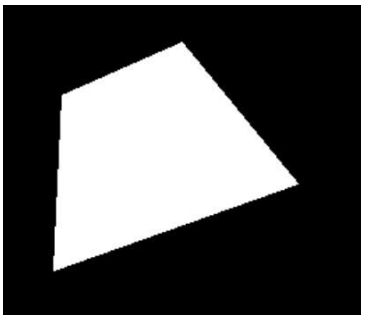
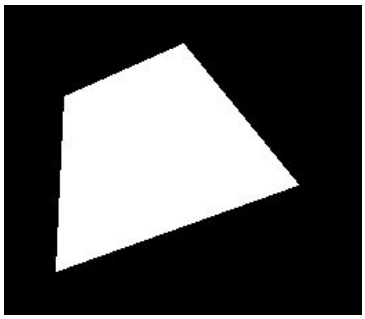




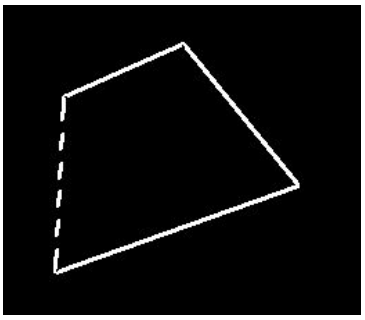
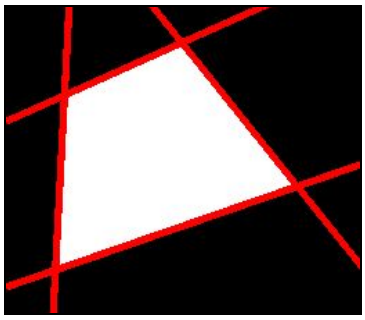
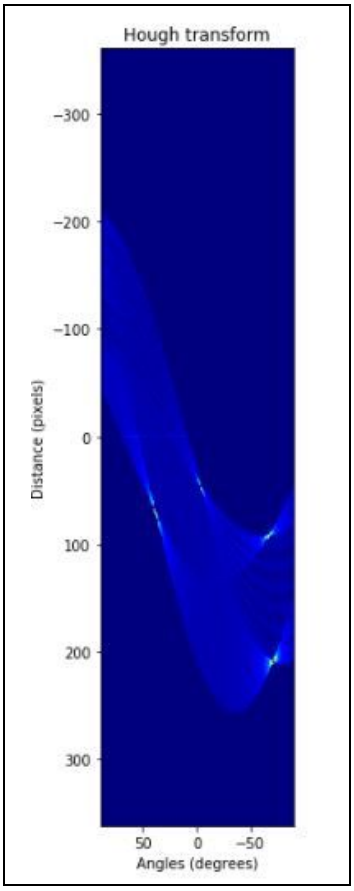
Local Maxima

---------- LM ----------

Val: 23 | [theta, rho]: [[[179, 309]]]
Val: 55 | [theta, rho]: [[[56, 312]]]
Val: 60 | [theta, rho]: [[[150, 465]]]
Val: 71 | [theta, rho]: [[[83, 388]]]
Val: 77 | [theta, rho]: [[[135, 410], [148, 408]]]
Val: 85 | [theta, rho]: [[[21, 166]]]
Val: 96 | [theta, rho]: [[[112, 344], [113, 345]]]
Val: 111 | [theta, rho]: [[[108, 467]]]
Val: 114 | [theta, rho]: [[[12, 253]]]

Results Analysis for test.bmp

Input	Greyscale	Gaussian
		

Canny Edge	Line Detection	Hough Parameter Space
		

Local Maxima

---------- LM ----------

Val: 12 | [theta, rho]: [[[42, 241], [45, 250]]]

Val: 14 | [theta, rho]: [[[0, 163], [1, 164], [1, 206], [2, 165], [2, 186], [3, 166], [3, 187], [3, 299], [5, 185]]]

Val: 28 | [theta, rho]: [[[31, 361], [32, 361], [33, 361], [34, 361]]]

Val: 89 | [theta, rho]: [[[93, 407]]]

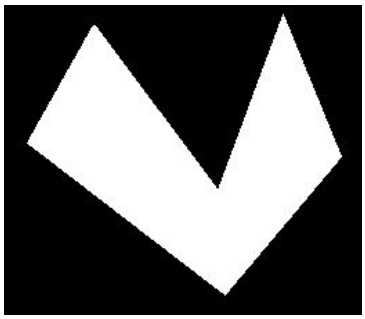
Val: 98 | [theta, rho]: [[[156, 453]]]

Val: 136 | [theta, rho]: [[[51, 432]]]

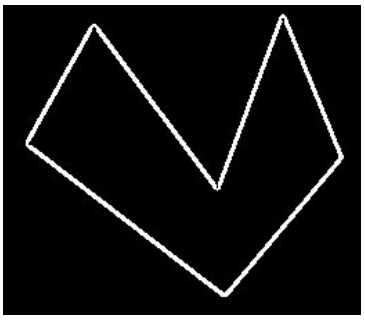
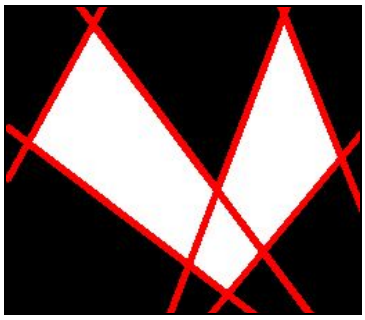
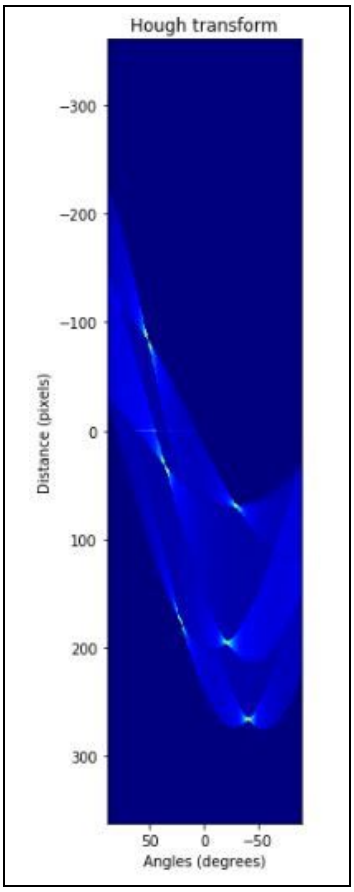
Val: 186 | [theta, rho]: [[[161, 569]]]

Results Analysis for test2.bmp

Input	Greyscale	Gaussian


Canny Edge	Line Detection	Hough Parameter Space

Local Maxima

---------- LM ----------

Val: 17 | [theta, rho]: [[[177, 402], [178, 401]]]

Val: 20 | [theta, rho]: [[[154, 486], [168, 464], [175, 517], [177, 512], [179, 507]]]

Val: 100 | [theta, rho]: [[[119, 431]]]

Val: 113 | [theta, rho]: [[[68, 538]]]

Val: 134 | [theta, rho]: [[[130, 628]]]

Val: 138 | [theta, rho]: [[[111, 557]]]

Val: 151 | [theta, rho]: [[[53, 392], [53, 393]]]

Val: 183 | [theta, rho]: [[[37, 277]]]
