CS 435 – Computational Photography
Assignment 5

In this assignment you will demonstrate your ability to apply several algorithms for the purpose of plane rectification. In particular:

1. The Hough Transform for line detection.
2. Planar homography.

Grading Scheme:

1. Theory Questions (11pts)
2. Edge Detection (5pts)
3. Hough Transform (20pts)
4. Line Selection (20pts)
5. Rectification (30pts)
6. Works on more than one sample image (10pts).
7. Report (4pts)

Theory Question(s)

1. Given a rotation of $30°$ (clockwise) and a translation of $d_x = -2, d_y = 1$ (in that order), what is the 2D homogenous transformation matrix. (2pts)

2. Apply the transformation matrix your found in the previous part to the following pixel locations to get the new set of locations (2pts):

$$P = \begin{bmatrix} -1 & 1 & 1 & -1 \\ -1 & -1 & 1 & 1 \end{bmatrix}$$

3. Plot/graph this before and after the transformation. Does it make sense? If you wish you may hand-draw this, take a photo of your drawing, and insert that image into your solution document. (2pts)

4. Using your set of four point correspondences from the previous two questions, what is the planar projection matrix (i.e. the homography)? You **cannot** use your prior knowledge (that the image was rotated and translated to make this matrix). Instead solve for its coefficients. (4pts)

5. Does your answer above make sense? Why or why not? Remember, that due to the $|h| = 1$ constraint, your planar projection matrix is only to scale (1pt).

Introduction

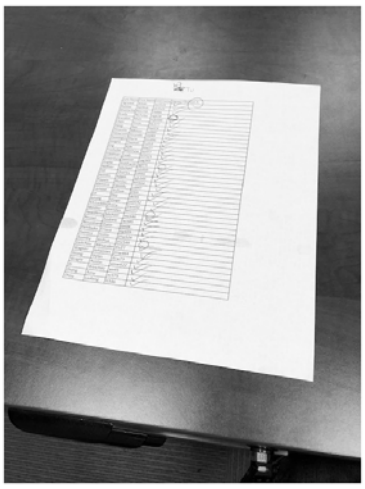
In this final assignment we will combine several concepts from earlier in the course with new concepts of line parameterization and image rectification. In particular, you will be developing a "scanner app".

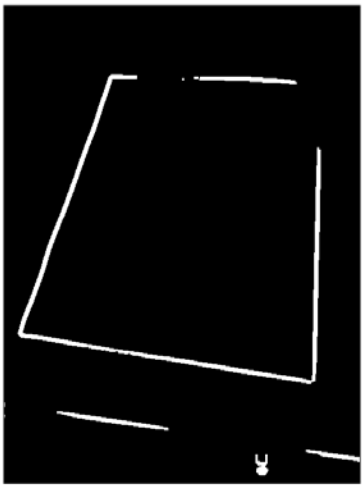
Throughout this you will use the image you took of a piece of paper.

Part 1: Edge Detection

First take your image and apply edge detection to it. Hopefully you did this successfully in HW2. If not, feel free to use a function in Matlab to do it.

For your report show the original image and the edge image.





Part 2: Hough Transform for Line Detection

Next, let's attempt to find lines in your edge image by applying a **Hough Transform** to the edge pixels.

Recall the form of a line in polar coordinates is:

$$\cos(\theta)\, x + \sin(\theta)\, y = r$$

For each edge pixel, vary $0 \leq \theta \leq 180$ and compute the associate r value. Quantize these (θ, r) pairs and "bin" them to create a 2D histogram. You can decide how "fine" you want your steps for each of the parameters to be. Visualize this 2D histogram as an image.

Part 3: Relevant Line Identification.

From the Hough transform you will likely be able to identify locations of local maxima. The difficulty is grabbing the correct subset to form the four lines of your piece of paper.

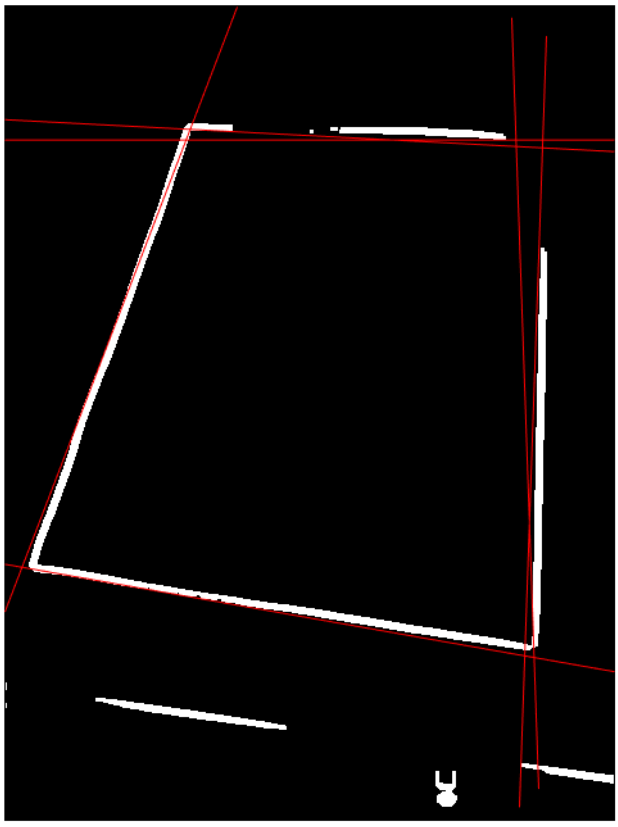
I'm going to let you think about how to do this yourself. Keep in mind that this should be automated such that it could be applied to another test image. To get you started, here's some ideas:

1. Find local maxima over some fixed window size.
2. Apply some sort of clustering.
3. Enforce some constraints based on the nature of the problem (in this case, finding the four edges of a plane/paper.
4. Set a threshold.

In your writeup explain how you selected potential lines (including citations, if necessary).

Finally, superimpose on your Hough transform these locations and draw the lines on your edge image.

NOTE: If you aren't able to do this automatically, just let us know and "manually" select them. That way you can move on to the last part.



Part 4: Line Intersections

Next you'll want to find the intersections of your lines to establish the four corners of your paper.

From the polar form of the line, $\cos(\theta)\, x + \sin(\theta)\, y = r$, you should be able to get the slope-intercept form of a line, $y = mx + b$.

Go through your potential lines and compute their intersections. Use this set of intersection points to determine the four corner points of your paper.

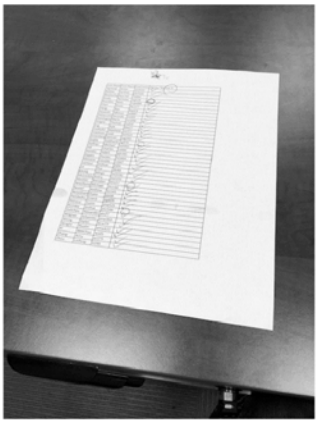
Part 5: Image Rectification

Now we should be able to rectify our paper! We will assume a standard 8.5×11 letter paper size. Based on this aspect ratio create a "blank" image (make it large enough to show details, likely similar size to the original image).

The four corners of this blank image should correspond to the four intersection points you discovered. Using these four correspondences, compute the *homography matrix.*

Once you have your homography matrix, go through each location in your new image and compute its corresponding location in the original image. Copy the pixel at that location to your new image.

Show your newly rectified image!



Part 6: Another Example!

Finally, we want to make sure that your implementation works on another example without having to change anything in your code.

Provide a second input and output image showing this.

Submission

1. Assignments must be submitted via Bd Learn
2. Submit a single compressed file (zip, tar, etc..) containing:
    a. A PDF file containing:
        i. Your answer to the theory question(s).
        ii. Your input and edge image for Part 1
        iii. Your 2D Hough Transform histogram image for Part 2
        iv. A description of how you extracted points from your Hough Transform and the selected lines superimposed on your edge image, for Part 3
        v. Your rectified image for Part 5
        vi. Your extra example, for Part 6
    b. A README text file (**not** Word or PDF) that explains
        i. Any additional features of your program
        ii. Name of your entry-point script
        iii. Any instructions on how to run your program.
    c. Your source files
    d. The chosen images that you are processing.