Assignment A9: Model Fitting

CS 5320/6320
Spring 2016

Assigned: 15 March 2016

Due: 6 April 2016

For this problem, handin a lab report A9.pdf (include name, date, assignment and class number in pdf) which develops and studies model fitting methods. These are to be applied to the hall images found in the class data/A9 directory; Gaussian noise should be added to the images and the affect on the line parameters determined. Propose a reasonable measure for this.

The goal is to develop a set of functions that can retrieve lines from the image. This involves a number of capabilties:

- Develop CS5320_total_LS, a total least squares function for line parameter estimation function (p. 295).
- Develop CS5320_Hough, a Hough transform, as well as some helper functions.
- (CS6320) Develop CS5320_line_segs, a function to return line segments (i.e., connected linear pixel sets such that for every pair of pixels in the line, there is a connected path between them that goes through other pixels in the line), based on the line finding results of other functions.
- (CS6320-optional) Develop CS5320_shapes, a function to return shapes (triangles and rectangles) found in an image.

The headers for the Matlab functions to develop are given below. As an example, Figure 1 shows the results of the Hough function applied to a hall image.

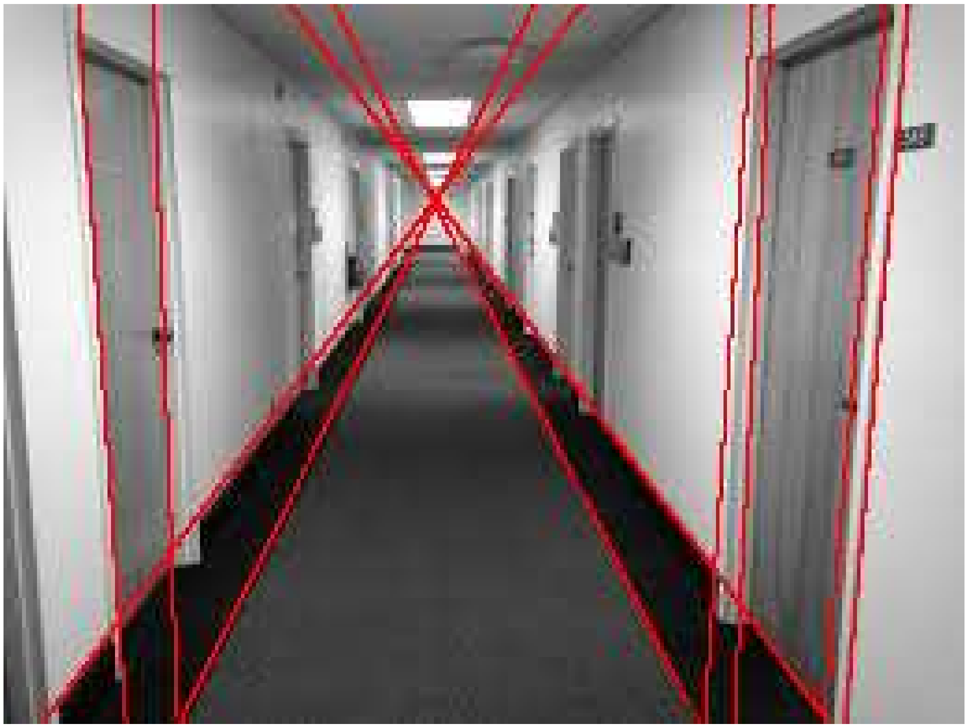


Figure 1: Hough Lines found in Hallway Image.

You should handin the report A9.pdf as well as the source code developed in the study. The code should conform to the style requested in the class materials.

In addition, please turn in a hardcopy of the report in class before the start of class on April 6, 2016.

Write a lab report in the format (please do not deviate from this format!) described in the course materials.

```
function [p,s] = CV_total_LS(x,y)
%
%CV_total_LS - total least squares method to fit best line to points
%    (Forsyth and Ponce page 335
%On input:
%     x (nx1 vector): x coordinates of points
%     y (nx1 vector): y coordinates of points
%On output:
```

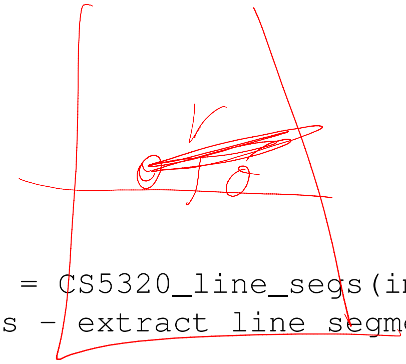
```
%     p (1x3 vector): coefficients of best fit line  ax + by + c = 0
%     s (float): error measure (sum of squares of distances
%                of points to line)
%Call:
%     [p1,s1] =  CV_total_LS([1,2,3],[1,2,3]);
%Author:
%     Tom Henderson
%     UU
%     Fall 2004
%

function [H,pts] = CS5320_Hough(imo)
% CS5320_Hough - Hough transform of image
% On input:
%     imo (mxn array): gray-level image
% On output:
%     H (rxt array): Hough accumulator array (r rho values; t theta
%     values)
%        r = [1:ceil(image diagonal)]; t = [1:180]
%     pts (rxt struct): contains points which contributed to line
%       pts(i,j).pts (kx2 array): k pixels (rows and cols)
% Call:
%     [H4,H4pts] = CS5320_Hough(double(hall4g));
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%

function lines = CS5320_Hough_lines(im,H,thresh)
% CS5320_Hough_lines - produce mask with lines
% On input:
%     im (mxn array): gray level image
%     H (rxt array): Hough accumulator (from CS5320_Hough)
% On output:
%     lines (mxn array): lines mask (gray is line number)
% Call:
%     lines = CS5320_Hough_lines(hall4g,H4,70);
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%

function lines = CS5320_Hough_draw_pts(im,H,pts,n,dr)
% CS5320_Hough_draw_pts - draws points which contributed to line
% On input:
%     im (mxn array): original image (or edge image)
%     H (rxt array): Hough accumulator array (see CS5320_Hough)
%     pts (rxt struct): record of points in Hough accumulator
%     n (int): number of lines to show (sorted by most points)
%     dr (Boolean): 1 means draw, 0 don’t
% On output:
%     lines (mxn array): mask of n lines
% Call:
%     linesp = CS5320_Hough_draw_pts(hall4g,H4,H4pts,10,1);
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%

function CS5320_plot_line(p,x1,x2,y1,y2)
% CS5320_plot_line - plots line given parameters
% On input:
%     p (1x3 vector): line parameters (p(1)x + p(2)y + p(3) = 0)
%     x1 (float): first x value
%     x2 (float): second x value
%       if x1 ˜= x2, then plot line from x1 to x2 (calculate y1 and
%       y2)
%     y1 (float): first y value
%     y2 (float): second y value
%       if x1==x2, then plot line from y1 to y2 (calculate x1 and x2)
% On output:
%     plot line
% Call:
%     Cs5320_plot_line(p,0,7,0,0);
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%
```



```
function segments = CS5320_line_segs(ime,Hpts,min_len)
% CS5320_line_segs - extract line segments from Hough info
% On input:
%     ime (mxn array): edge image (e.g., output of edge)
%     Hpts (txr array): Hough points array (see Cs5320_Hough)
%     min_len (int): minimum segment length
% On output:
%     segments (struct vector): segment info
%       (s).pts (kx2 array): row,col points in segment
%       (s).rho (int): rho parameter of line
%       (s).theta (float): theta parameter of line
%       (s).endpt1 (1x2 vector): one endpt of segment
%       (s).endpt2 (1x2 vector): other endpt of segment
% Call:
%     As = CS5320_line_segs(A,HApts,20);
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%

function shapes = CS5320_shapes(segs)
% CS5320_shapes - find triangles and rectangles
% On input:
%     segs (segs data structure): segments (see CS5320_line_segs)
% On output:
%     shapes (struct vector): shape info
%      (s).segs (1xk vector): segment indexes of shape
%            currently either 3 (triangle) or 4 (rectangle)
% Call:
%     Tt = CS5320_shapes(square);
% Author:
%     T. Henderson
%     UU
%     Spring 2016
%
```