Exercise in Robust Model Fitting

Henrik Aanæs, haa@imm.dtu.dk

October 26, 2010

This is the exercise in robust model fitting for course 02501. This exercise covers/illustrates the material of Chapter four of my lecture notes. In this exercise you are expected to perform line extraction via the Hough transform, as well do line fitting via the Ransac algorithm.

1 Hough Transform

Here you should extract lines from the image BookImage.jpg, via the Hough transform. The image should be found together with this exercise. Perform the following steps:

1. Load the image and extract lines from it. Here the MatLab function edge should be of help. try different edge detection algorithms - what is the difference between them?

2. Compute the Hough space from the extracted lines. Use the function [H, T, R] = hough(BW). What is interpretation of the parameters H,T and R?

3. Display the Hough space H.

4. Apply the following transformation to the hough space and display it

   ```
   H2=H/max(H(:));
   H2=H2.^ 0.5;
   ```

   What does this transformation do, and why?

5. Extract the peaks from the Hough space H, using houghpeaks.

6. In the displayed Hough space annotate the positions of the extracted peaks.

7. Display the original image, BookImage.jpg, and annotate the extracted lines in it. Use the function DrawImageLine, which should be found together with this exercise text.

8. What is the effect of using different edge detection algorithms as input to the Hough line extraction?

2 Ransac

Here you should estimate a line to a data set consisting of inliers with noise and outliers. Such data is generated by the supplied function x=RanLine(In,Out). It is recommended that you do so in the following steps.

1. Make a function that estimates a line, in homogeneous coordinates, given two points.

2. Make a function that determines *if* a given 2D point is inlier or outlier wrt. a given line. The threshold should be supplied as parameter to this function, such that it can easily be tuned latter.

3. Make a function that calculates the consensus, i.e. number of inliers, for a line wrt. to a set of points.

4. Make a function that randomly draws two of n 2D points.

5. Assemble the functions made above to a working Ransac algorithm for estimating lines. The number of iterations is set manually.

6. Experiment with the algorithm, what is a good threshold for distinguishing between inliers and outliers?