CS-512 – Assignment 4 (4%)

Camera calibration

Due by: November 14, 2017

Review questions

Answer the following questions. Make sure that your answers are concise. In questions requiring explanations, make sure your explanations are brief.

1. Robust estimation and segmentation
   (a) Explain what are outliers, and is the fundamental problem associated with them when fitting a model.
   (b) Write the objective function that is used for robust estimation and explain its difference from the standard least squares objective function.
   (c) Write the Geman-McClure function for robust estimation and explain its advantage. Explain how the bandwidth parameter σ can be adjusted in an iterative manner.
   (d) Explain the principle of the RANSAC algorithm. Should the number of points drawn at each attempt be large or small? Explain why?
   (e) Explain the parameters of the RANSAC algorithm. Write the formula for estimating the number of trials.
   (f) Explain the objective of image segmentation. Explain the differences between agglomerative (merge) and divisive (split) approaches.
   (g) Explain the k-means and mixture of Gaussians segmentation algorithms.
   (h) Explain the mean-shift algorithm for segmentation.

2. Camera calibration
   (a) Given the projection equation $p = MP$ explain the problems of: forward projection, calibration, and reconstruction. Which problem is the easiest? Which is the most difficult?
   (b) Explain what is the necessary input for camera calibration.
   (c) Explain the steps in the non-coplanar calibration algorithm.
   (d) Given a known projection matrix with rows $(1, 2, 3, 4)(1, 0, 3, 4)(1, 1, 1, 1)$ and a world point $P_i = (1, 2, 3)$ what are the 2D image coordinates of P_i after projecting it. Make sure to convert P_i to homogeneous coordinates before projecting it.
   (e) Given the corresponding world-image points: $(1, 2, 3) \leftrightarrow (100, 200)$ write the first two lines of the matrix that needs to be formed for solving for the unknown projection matrix M.
   (f) What is the minimal number of points that is necessary to be able to find a unique solution for M. How is the solution obtained?

(g) Explain the principal that is used to extract the unknown camera parameters from the projection matrix M.

(h) Explain how to compute the quality of the projection matrix M estimate.

(i) Explain the principal of planar camera calibration. How does planar camera calibration differ from non-coplanar one?

(j) Explain the difference between the homography (2D projective map) H and the projection matrix M. What is the assumption that is used to make sure we deal with homography matrices?

Programming questions

In this assignment you need to implement a camera calibration algorithm under the assumption of noisy data. You need to implement either planar calibration or non-coplanar calibration. Note: you only need to implement one calibration method. Robust estimation through RANSAC should be used to eliminate outliers. The program should satisfy the following specifications:

1. Write a program to extract feature points from the calibration target and show them on the image. You may use the openCV functions to do so (e.g. `cvFindChessboardCorners`, `cvFindCornerSubPix`, `cvDrawChessboardCorners`). Alternatively write a program that allows you to interactively mark the points on the image. Save the image points detected and/or manually entered in a file.

2. Write a second program to compute camera parameters as described below that uses the point files produced by the first program.

   - Program arguments: In the case of non-planar calibration the program argument is the name of a single point correspondence file. In the case of planar calibration the program arguments are the names of 3 point correspondence files specifying correspondence in 3 different views. A point correspondence file is a text file containing in each row a pair of corresponding points (3D-2D) as real numbers separated by space. The first 3 numbers in each row give the x, y, z coordinates of a 3D world point whereas the last 2 numbers give the x, y coordinates of the corresponding 2D image point.
   - After completing the calibration process, the program should display the intrinsic and extrinsic parameters of the camera as determined by the calibration process. The program should compute and display the mean square error between the known and computed position of the image points. The computed position should be obtained by using the estimated camera parameters to project the 3D points onto the image plane. You do not need to address radial lens distortion in your calibration.
   - Write the programs using Python with OpenCV. You may not directly use the openCV calibration function (e.g. `cvCalibrateCamera`). However, you may use this function for verification.

3. Implement the RANSAC algorithm for robust estimation. Your implementation of the RANSAC algorithm should include automatic estimation of the number of draws and of the probability that a data point is an inlier. The final values of these estimates should be displayed by the program. In your estimation of these values, assume a desired probability of 0.99 that at least one of the draws is free from outliers. Set a maximum number of draws that can be performed. When testing the program on noisy data you will note that RANSAC is not handling well one of the provided cases. Explain the reason for RANSAC not being able to handle this case properly. Parameters used in the RANSAC algorithm should be read from a text file named 'RANSAC.config'.

4. Data files for testing the calibration are available on the course website. In addition to testing your program with these files, make sure to test it with real images of a calibration target that you generate.

5. Submission should be done as described in assignment 1.