ETH Zürich
Computer Science Department

Prof. Marc Pollefeys
Prof. Luc Van Gool
HS 2013

Computer Vision

Exercise 3

Hand-out: 08-10-2015
Hand-in: 15-10-2013 (13:00hs)

Objective:

The goal of this assignment is to estimate the epipolar geometry between two related views. The fundamental matrix for two uncalibrated cameras is estimated by implementing the eight-point algorithm. For the calibrated case, you will estimate the essential matrix and compute the relative camera poses. All coding should be done in Matlab.

Documentation:

Inside the `material.zip` file, the slides can be found in the `doc/` folder. In addition, there is a paper and a book chapter from Hartley and Zisserman, it will complement the information provided. Please take a look in the `README.txt` files.

3.1 Image capturing (10%)

Capture two images from a static 3D scene from different viewpoints. The images should have some overlap such that point correspondences can be established. Undistort your images to make straight lines in the scene appear straight in the image. This can be achieved using the Bouget's toolbox used in the previous exercise.

3.2 Fundamental matrix (40%)

Compute the fundamental matrix F that relates the two images by implementing the eight-point algorithm. Make sure you include the following in your implementation:

- Your eight-point algorithm should be a homogeneous least-squares solution that includes more than 8 point correspondences. Select the point correspondences by clicking manually on the two images.
- Do not forget to normalize the points before applying the eight-point algorithm (Scale the points so that their root-mean-squared distance is $\sqrt{2}$).
- Enforce the singularity constraint. Draw the epipolar lines for both, the non-singular fundamental matrix F and the fundamental matrix $\hat{F}$ with enforced singularity constraint. Also show the epipole of F and $\hat{F}$ in both images, which are the right and left null-vectors of the fundamental matrix.

3.3 Essential matrix (20%)

The essential matrix can be computed with known camera calibration. Implement another eight-point algorithm to compute the essential matrix. Use the same set of point correspondences from the previous section. Make sure you include the following in your implementation:

- Similar to the eight-point algorithm for the fundamental matrix, your eight-point algorithm for the essential matrix should be a homogeneous least-square solution that includes more than 8 point correspondences.
- Make sure that the normalized image coordinates are used.
- The constraint of the first two singular values must be equal and the third one is zero has to be enforced.

3.4 Camera matrix (30%)

The camera matrix $P' = [R|t]$ is made up of the relative transformation (R, t) between the second and the first camera. The first camera matrix $P = [I|0]$ is taken to be the origin of the camera coordinate system. (R, t) can be computed from the essential matrix and four possible solutions exist. Only one of these four solutions is correct.

a) Implement the code to compute all the four possible solutions for the camera matrix P' from the essential matrix obtained in the previous section. You should normalize t to unit length and make sure that R follows the right hand coordinate system.

b) Choose the correct camera matrix P' from the four possible solutions. (Hint: This can be done by checking any one of the triangulated 3D points using the triangulation function provided.)

3.5 Non-linear optimization (Bonus: +10%)

With the camera matrix P' found from the previous section, use the triangulation function provided to triangulate the 3D points X_i for all the selected point correspondences $x_i \leftrightarrow x'_i$. Next, improve the accuracy of the camera matrix P' and all the triangulated 3D points X_i with a non-linear least square optimization process that minimizes the cost function

$$\sum_i d(x_i, PX_i)^2 + d(x'_i, P'X_i)^2 \tag{1}$$

where $d(.,.)$ is the Euclidean distance between the true coordinates x_i and the reprojected coodinates PX_i. The "fminsearch" function in Matlab can be used for optimization. Report the decrease of your reprojection error after the optimization. Plot, on the same figure, the camera poses and 3D points before and after optimization (use different colors for before and after optimization).

3.6 Hand in:

Write a short report explaining the main steps of your implementations and discuss the results. Be sure to include

- Images before and after undistortion.
- Images overlaid with plots of epipolar geometry.
- Results of the estimated fundamental matrix, essential matrix, rotation and translation of the camera.

Send the report together with your source code to **pablo.speciale@inf.ethz.ch**. Your source code should include

- The intrinsics of your camera, used to solve the assignment, hard coded in the main.m files.
- The two undistorted images with the clicked point correspondences stored as a .mat file.

3.7 References:

[1] `http://www.vision.caltech.edu/bouguetj/calib_doc/index.html`