Exercise 9: Geometry Constrained Feature Matching

02504 Computer vision

Morten R. Hannemose, mohan@dtu.dk, DTU Compute

April 5, 2024

The Eight Point Algorithm

Here you will write a function that uses the eight point algorithm for fundamental matrix estimation. This should be a slight modification of your algorithm for homography estimation. As for the homography estimation the data points should also be normalized.

Find the fundamental matrix that you computed for the images in `TwoImageData.npy` in week 3 and denote it `Ftrue`.

Exercise 9.1

Implement a function `Fest_8point` that estimates a fundamental matrix from eight or more point correspondences, using the linear algorithm.

Test your function using the points in `Fest_test.npy`. Check that your estimated fundamental matrix is identical to `Ftrue` up to scale and numerical error.

You can load the file using `np.load('Fest_test.npy', allow_pickle=True).item()`.

Feature Matching

Exercise 9.2

Download `TwoImageData.npy` (not the car version). As you did last week, find features in both images and match them. However, this time do not filter matches by the ratio test, only use cross checking as done by `cv2.BFMatcher_create(crossCheck=True)`.

Visualize the result and confirm that it looks reasonable compared to your expectations.

Fundamental matrix estimation via RANSAC

Make a copy of your RANSAC algorithm that fits straight lines, and modify it to fit fundamental matrices instead.

Exercise 9.3

Sample eight random matches. This can be done with the following code
`np.random.choice(matches, 8, replace=False)`.

Use your function `Fest_8point` to estimate the fundamental matrix from these eight matches.

Write a function `SampsonsDistance(F, p1, p2)` that computes Sampson's distance.

Set points to inliers if their Sampson's distance is less than $3.84 \cdot 3\hat{}2$. Explain where this value comes from.

Repeat the above steps for the a fixed number of iterations, such as 200.

Finally, use `Fest_8point` to estimate the final fundamental matrix using all inliers of the best model.

Run your algorithm on the the images from `TwoImageData.npy`. Compare your estimated `F` to `Ftrue` using the following code:
`(F*Ftrue).sum() / (np.linalg.norm(F)*np.linalg.norm(Ftrue))`.
Explain what this code does.

Exercise 9.4

Find the images you captured last week or capture new ones. Match SIFT features between these images using cross checking.

Estimate the fundamental matrix between these images using your RANSAC algorithm.

Comment on how well the fundamental matrix acts as a regularizer on which matches are used.