The University of Melbourne
School of Computing and Information Systems
COMP90086 Computer Vision, 2022 Semester 2

Assignment 3: Fundamental Matrix Calculation

Due: 7pm, 7 Oct 2022
Submission: Source code in Jupyter Notebook with copious comments
Marks: The assignment will be marked out of 8 points, and will contribute 8% of your total mark.

Introduction

In this project you will implement the calculation of a Fundamental matrix using your own algorithms. You can use the keypoint detection and matching approach used in the week 8 Workshop, as well as code for drawing lines on images, but the implementation of the Fundamental Matrix calculation must be your own work.

Coding tasks

Your tasks are:

1. Find keypoints and correspondences between two images
2. To implement the 8 point algorithm you will need to:
    a. Shift and scale the pixel coordinates
    b. Compute the design matrix from sets of (at least) 8 points
    c. Perform an SVD of the design matrix to find its null space (you can use a library function for the SVD)
    d. Compose the draft fundamental matrix F
    e. Perform an SVD of the draft fundamental matrix, set the smallest singular value to zero and reassemble so that F now has determinant = 0
    f. Calculate which correspondences are inliers and which are outliers using this F. You will need to allow for some error because the keypoint coordinates won't lie exactly on the epipolar line so you will need to allow for 1-2 pixels error for a correct correspondence. Also remember to take into account any scaling you applied at step a.
    g. Wrap steps b-f in a RANSAC loop that runs enough times that you have a probability > 99% of finding 8 inliers and computing a good quality F.
    h. Re-estimate F using all the inliers.
    i. Compute F in terms of the original pixel coordinates (ie undo the effects of step a).
3. Randomly sample ten keypoint pairs from correspondences you have detected as inliers to your best F estimate and display these on the images, together with their epipolar lines.

If you make this a separate cell in your jupyter notebook, you can run the cell more than once to see different random samplings.

Report

Your report should describe why you need each of these steps and any design choices you have made along with numerical quantities, e.g. choice of error tolerance at step f, or the calculation needed to determine the number of RANSAC iterations at step g.

You should also discuss whether there are images that work better than others in the dataset and why.

Dataset

The dataset is the kusvod2 dataset. Image files are of the form <something>A.png and <something>B.png for each image pair.

Using Library Code

You may use library code for everything except the implementation of the 8 point algorithm which you must implement yourself. This means that you can use library implementations of keypoint detection and correspondence (e.g. SIFT or FAST and BRIEF) as well as implementations of SVD or other linear algebra operations.

Submission

You should make two submissions on the LMS: your code and a short written report explaining your method and results. Your report should be no more than 1500. Submission will be made via the Canvas LMS. Please submit your code and written report separately under the **Assignment 3: Code** and the **Assignment 3: Report** links on Canvas.

- Your code submission should include the Jupyter Notebook (please use the provided template) with your code and any additional files we will need to run your code. You do not need to include the provided images in your submission.
- Your written report should be a .pdf with your answers to each of the questions. The report should address the questions posed in this assignment and include any images, diagrams, or tables required by the question.

Evaluation

Your submission will be marked on the correctness of your code/method, including the quality and efficiency of your code. You should use built-in Python functions where appropriate and use descriptive variable names. Your written report should clearly explain your approach and any experimentation used to produce your results, and include all of the specific outputs required by the question (e.g., images, diagrams, tables, or responses to sub-questions).

Late submission

The submission mechanism will stay open for one week after the submission deadline. Late submissions will be penalised at 10% of the total possible mark per 24-hour period after the original deadline. Submissions will be closed 7 days (168 hours) after the published assignment deadline, and no further submissions will be accepted after this point.

Updates to the assignment specifications

If any changes or clarifications are made to the project specification, these will be posted on the LMS.

Academic misconduct

You are welcome — indeed encouraged — to collaborate with your peers in terms of the conceptualisation and framing of the problem. For example, we encourage you to discuss what the assignment specification is asking you to do, or what you would need to implement to be able to respond to a question.

However, sharing materials — for example, showing other students your code or colluding in writing responses to questions — or plagiarising existing code or material will be considered cheating. Your submission must be your own original, individual work. We will invoke University’s Academic policy (**http://academichonesty.unimelb.edu.au/policy.html**) where inappropriate levels of plagiarism or collusion are deemed to have taken place.