

## COMP 558: Assignment 3

Available: Monday, November 15th, 2021

Due Date: Thursday, December 2nd, 2021 (before midnight) via mycourses.

## Introduction

This assignment covers material related to SVD, least-squares estimation, stereo and epipolar geometry. The relevant lectures and slides are from lectures 17-21. The questions concern camera calibration and in particular recovery of the Fundamental matrix, and further applications to stereo and stereo reconstruction. This particular assignment is a bit research oriented in that for some of the questions the answers might be a bit open-ended. For these we want to see what you come up with, and how you present your results and discuss them.

Please use the mycourses discussion board forum for assignment 3, for any assignment related questions. This will help us reach the entire class when we respond. However, follow appropriate protocol, e.g., do not reveal the answer to a particular question or ask if your proposed solution is correct. Save these kinds of questions for one-on-one interactions during office hours. Do not post pictures of your solutions because these could give ideas to the rest of the class. You are free to discuss the general aspects of questions with each other, but not to the point where you are revealing answers or sharing solutions. This is an individual assignment and the solutions that you submit must reflect your own work and must be written by you. You are not permitted to submit code or text that you have copied from one another or from any third party sources, including internet sites, such as the Matlab file exchange forum. We expect that you will write your code. By submitting this assignment you agree to abide by these rules. Keep an eye on the discussion board where we might post hints or reply to questions that come up!

## Instructions

- All submissions should be made to the assignment 3 folder.
- For this assignment you should submit a single .zip file containing 1) a PDF file explaining your solution to each part in some detail, and providing illustrative examples and/or figures as you need them, and discussing your solutions 2) your Matlab code, with comments, so that we can run it and 3) if you like, sample movies or images illustrating your reconstruction results.
- In order to receive full points, your solution code must be properly commented, and you must provide a clear and concise description of your computed results in the PDF. The TAs have limited time and will spend at most 20-30 minutes grading each submission.

**Late assignment policy:** Late assignments will be accepted up to only 3 days late and will be penalized by 10% of the total marks, per day late. For example, an assignment that is submitted 2 days late and that receives a grade of 90/100 would then receive an 18 point penalty. Submitting one minute after midnight on the due date means it is 1 day late (not a fraction of a day), and the same for the second day.

## Question 1: Fundamental Matrix Estimation [30 marks]

We have provided two stereo pairs for this assignment. You will be using these pairs to estimate the Fundamental matrix  $\mathbf{F}$  using the 8-point algorithm. Start with one pair and manually identify  $n$  corresponding pixel pairs, where initially  $n = 8$ . Be sure to choose correspondences that appear to be visually unique, i.e., they appear to be projections of a unique 3D scene point in the two camera images. You can get pixel coordinates just by using *imshow* and appropriate menu options in the image viewer. In Matlab there are also ways to interactively click and record image pixel locations.

Recall from Lecture 21 and the associated slides, that estimating  $\mathbf{F}$  robustly requires a number of sequential steps: 1) data normalization, 2) solving a least squares problem to estimate the entries of  $\mathbf{F}_{norm}$ , 3) mapping these entries to the entries of  $\mathbf{F}$  via a de-normalization step, and 4) enforcing a rank = 2 constraint on  $\mathbf{F}$ . When you are done with this part you should have a  $3 \times 3$  matrix for a particular image pair. We will check your code to make sure you have completed these steps.

In your PDF for this part you should describe your steps, and you should output your  $\mathbf{F}$  matrix for each image pair. You should also demonstrate that the  $\mathbf{F}$  matrix has rank 2 (you can use Matlab's *rank* function).

## Question 2: Fundamental Matrix Refinement - RANSAC [20 marks]

You will now build on your solution to question 1, by considering a longer list of corresponding pixel pairs, e.g.,  $n = 20$  or larger. To make this part easier for you we have provided some helper code which uses built in Matlab functions to find SURF features, and then generates the indices of potentially matching pairs. You can use this helper code to generate your list of candidate matches.

Now, modify your solution to question 1 by implementing a RANSAC algorithm, which works as follows:

1. Pick 8 pixel pairs randomly from the list of candidate pairs.
2. Estimate  $\mathbf{F}$ .
3. Determine how many of the remaining (unused) pixel pairs are in the consensus set of your estimate. For this you will have to develop a method to determine when a new pair is or is not in the consensus set.
4. If your consensus set is above a certain threshold, terminate and keep  $\mathbf{F}$ , but if not, start again.

When you are done with this part you should in principle have a more robust estimate of  $\mathbf{F}$  in that it should not be too sensitive to outliers (poor choices for pixel pair correspondences). For example, Slide 14 in Lecture 21 shows an example of SIFT feature based automatic pixel pair selection, where not all candidate matches are “good” ones. The idea of a RANSAC based refinement is to reject outliers when estimating  $\mathbf{F}$ .

In your PDF for this part you should describe your steps, and you should output your  $\mathbf{F}$  matrix for each image pair. You should also demonstrate that the  $\mathbf{F}$  matrix has rank 2 (you can use Matlab's *rank* function). Finally, you should compare these results with the ones you obtained in question 1.

### Question 3: Rectification [20 marks]

We will now use the estimated  $\mathbf{F}$  from question 2 to rectify each image pair. The process for doing this was outlined in class and it exploited the construction of a particular homography (Lecture 21, slide 22 onwards). Briefly, you will first need to find the two epipoles  $\mathbf{e}_1, \mathbf{e}_2$  as left and right null spaces of  $\mathbf{F}$ . Then you will define two homographies  $\mathbf{H}_1, \mathbf{H}_2$  each of which will map the corresponding epipole to the point at infinity in the  $x$  direction. Then you will directly apply these homographies to the two original images in the pair to generate warped images that are rectified. The quality of the rectification will give you a sense of the quality of your fundamental matrix and epipole estimates. Since you are applying homographies to the original images you will expect some stretching and distortion. Deal with boundary effects in a manner that is appropriate.

In your PDF you should describe the steps that you took, and then display each rectified image pair, and discuss the quality of your results.

### Question 4: Reconstruction (Original Problem) [20 marks]

Now assume that the matrices of “intrinsic” parameters of the two images in each pair are:

$$M_r = M_l = \begin{bmatrix} f/s_x & 0 & p_x \\ 0 & f/s_y & p_y \\ 0 & 0 & 1 \end{bmatrix},$$

where the pixels sizes are  $s_x = s_y = 31.2 \times 10^{-6}\text{m}$ . Here  $s_x = 1/m_x, s_y = 1/m_y$ . You can assume that the image center (the principal point)  $(p_x, p_y)$  is actually located in the center of the image. For stereo pair 1, the focal length  $f = 31\text{mm}$ , and for stereo pair 2, the focal length  $f = 32\text{mm}$ . Implement Algorithm **TRIANG** (Trucco and Verri, section 7.4.1, pages 162-163, descriptions attached) and use it to recover the depth of several corresponding pixel pairs chosen manually, in each stereo pair. For each stereo pair, choose your corresponding pixel pairs so that they belong to at least 5 different objects, so that you can recover at least 5 different depth levels.

In your PDF describe the steps you took. You are free to be creative when you display the resulting depths on the input images, for each pixel pair you have used. For example, you can try to use color coding for depth, or you could show depth values at chosen pixels using text. You will know qualitatively that your results are good if you pick pixels which correspond to 3D scene points which lie on planar surfaces, where you can tell by inspection what the relative ordering (in depth) of these 3D scene points is.

### Question 5: Dense Reconstruction - Disparity [10 marks]

This last part is somewhat open-ended and you can follow different paths to obtain a solution. Just be sure to clearly describe what you did. For this part we will consider the rectified images you generated in Question 3, as if the original camera axes were aligned, to obtain a depth map. We observed in class that in such a scenario the epipolar lines lie along horizontal axes, and that the binocular disparity  $d \equiv x_l - x_r$  is inversely proportional to the depth of the corresponding scene point, i.e.,  $d = fT_x/Z_0$ . Qualitatively, therefore, the depth of a 3D scene point is inverseley proportional to the disparity.

Referring to slide 31 from Lecture 21, choose  $T_x$  to be a parameter, and use the value of  $f$  we have given above. Now, using a 1D Lucas-Kanade type method (or a correlation based method,

or any appropriate local feature matching method) along a certain row, generate matches between pixels in the left and right images. Once you have these matches, you also have a disparity map. Display the resulting disparity map and comment on how well it serves to reconstruct the depth of points in 3D scene, i.e., discuss where it appears to succeed and where it fails, and why. We are not looking for perfect results, but rather, for a presentation, discussion and critical assessment of what you did.