CS59300CVD Assignment 4

Due date: Sunday, Nov. 24, 11:59 PM

This assignment was adapted from Svetlana Lazebnik's course on computer vision.

Homework Policy: Each student should write up **their solutions independently, no copying of any form is allowed.** If you have used online resources to help with answering the questions, but have come up with your own answers please properly reference the resources. You are allowed to use online resources as long as the answer is your own and properly acknowledge it. Please refer to the course website for policy on academic honesty. Finally, You MUST USE the provided LaTeXtemplate to typeset the report.

1 Assignment Details

1.1 Part 1: Fundamental Matrix Estimation, Camera Calibration, Triangulation

Objective: In this assignment, you will first implement the **fundamental matrix estimation** algorithm in Lectures 29-30. You will then implement the **camera calibration** and the **triangulation** in Lectures 26-28. The image pairs are provided in the starter code.

Method description:

1. **Fundamental matrix.** For the `library` and `lab` image pairs, find the fundamental matrices using the provided 2D matches in `X_matches.txt`. Implement the algorithms for estimating the unnormalized $\boldsymbol{F}$ and the normalized $\tilde{\boldsymbol{F}}$.

2. **Camera calibration.** For the `lab` image pair, find their camera projection matrices $\boldsymbol{P}$ respectively. Use the provided 2D matches in `X_matches.txt` and the 3D point coordinates in `X_3d.txt`.

3. **Triangulation.** For the `library` and `lab` image pairs, use linear least squares to triangulate the 3D position of each matching pair of 2D points given the two camera projection matrices. Then project the 3D points to 2D again using the projection matrices. NOTE: For the `lab` image pair, use the camera projection matrices you derived from **camera calibration**. For the `library` image pair, use the camera projection matrices provided in `library1_camera.txt` and `library2_camera.txt`.

1.2 Part 2: Affine Factorization

Objective: This part of the assignment aims to implement the Tomasi and Kanade affine structure from the motion method. You will be working with Carlo Tomasi's 101-frame hotel sequence provided in the starter code. Please refer to the "Structure from Motion" lecture slide from Lectures 30-31.

Method description:

1. Load the match measurements from the data file `measurement_matrix.txt`. Normalize the raw data by subtracting the centroid coordinates of image points of each view.

2. Derive an approximation of the structure matrix $\boldsymbol{S}$ and the motion matrix $\boldsymbol{M}$. Find the matrix $\boldsymbol{Q}$ to eliminate the affine ambiguity.

2 What to include in the report? (10 points)

2.1 Part 1

- Mathematically describe the formulation of the estimation algorithm for the $\boldsymbol{F}$, e.g., the inputs and the outputs. Clearly define all introduced notations. (1 point)
- Show the fundamental matrices. Please scale $\boldsymbol{F}$ and $\tilde{F}$ so that $F_{33} = \tilde{F}_{33} = 1.0$. Use the provided `visualize_fundamental` to visualize the epipolar lines. Use the provided `get_residual` to report the MSE of the matching points and their corresponding epipolar lines. What are your observations? (2 points)
- Mathematically describe the formulation of the camera calibration algorithm for the projection matrix $\boldsymbol{P}$, e.g., the inputs and the outputs. Clearly define all introduced notations. (1 point)
- Show the projection matrices. Use the provided `evaluate_points` to compute the MSE error of the projected 2D points. (1 point)
- Show the MSE error of the 3D points prediction from your triangulation. Show the MSE errors of the projected 2D points. (1 point)

2.2 Part 2

- Mathematically describe how you acquire structure and motion matrices, e.g. the inputs and the outputs. Clearly define all introduced notations. (2 points)
- Show the side-by-side comparisons of $\boldsymbol{S}$ before and after eliminating affine ambiguity with $\boldsymbol{Q}$ using `visualize_structure`. Pick 3 frames and show the side-by-side comparisons between the observed feature points and the estimated projected points using `visualize_keypoints_comparison`. Report their MSE respectively. Plot the MSE of each frame using `visualize_errors`. (2 points)

While code is not explicitly graded. Poorly commented or written code will result in points deducted from the previous three parts. For example, if the grader cannot understand what was implemented.

What to include in the code?

The code should contain a README.md file that describes how to run and reproduce your result. This includes the packages installed and their versions. We expected adequately commented code. To help with readability, please follow a style guide and use a linter, e.g., PEP8 is usually a good choice.

How to submit?

- You should have received an e-mail with the link to access Gradescope. If you haven't, let the TAs know.
- For your pdf file, use the naming convention `username_hw#.pdf`. For example, your TA with username *alice* would name her pdf file for HW4 as `alice_hw4.pdf`.

- Use the provided LaTeX template to typeset the assignment by editing the tex files under `student_response/`.
- After uploading your submission to Gradescope, mark each page to identify which question is answered on the page. (Gradescope will facilitate this.)
- For your code, use the naming convention `username_hw#.zip`. The code can be submitted via Gradescope under **assignment4 programming**.