AI Winter School
18^{th} December 2019

Dr. Danda Pani Paudel
Dr. Ajad Chhatkuli

Exercise - Structure from Motion

Hand-out: 18-12-2019
Hand-in: 30-01-2020 23:29
ajad.chhatkuli,paudel@vision.ee.ethz.ch

Objective:

In this exercise you create a very simple structure from motion pipeline. You are provided a python script to obtain a basic reconstruction and to build upon.

Source: Drive folder: Day9-3DVision

Feature extraction and epipolar geometry (40%)

To start the pipeline the first thing to do is to extract features in every image. Then select two images that have a large baseline but at the same time a big overlap so that there are enough feature matches. The larger the baseline, the better the triangulation of the feature points.

- In the code framework 5 images are included. For this step take any tow images of the sequence. Extract KAZE features and match them.
- Run either 8-point or 5-point RANSAC to compute the inlier-set and then compute the essential matrix. Decompose the essential matrix in to R and t and create the projection matrix for the second view assuming that the first one is $[I \mid 0]$.
- Triangulate the matched inlier features with the existing code and store their 3D positions. Now you have 3D-2D point correspondences for the inlier features.

Triangulation and adding new views (40%)

To add additional views of the scene you can now match 2D features in the new view and one of the existing views. These matches represent now also 3D-2D correspondences, the 2D point from the new image and the 3D from the 3D-2D correspondence of the existing view.

To find the pose of the new camera relative to the scene (the P matrix) you can use the 6-point DLT algorithm, making use of the function *solvePnPRansac*. To filter out wrong matches, use a 6-point RANSAC, that uses the reprojection error as error measure.

7.1 Plotting (20%)

To analyze your results, triangulate all inlier matches from the previous task and plot them in 3D together with the inliers from the first task. Also visualize the 3D points reconstructed from all images.

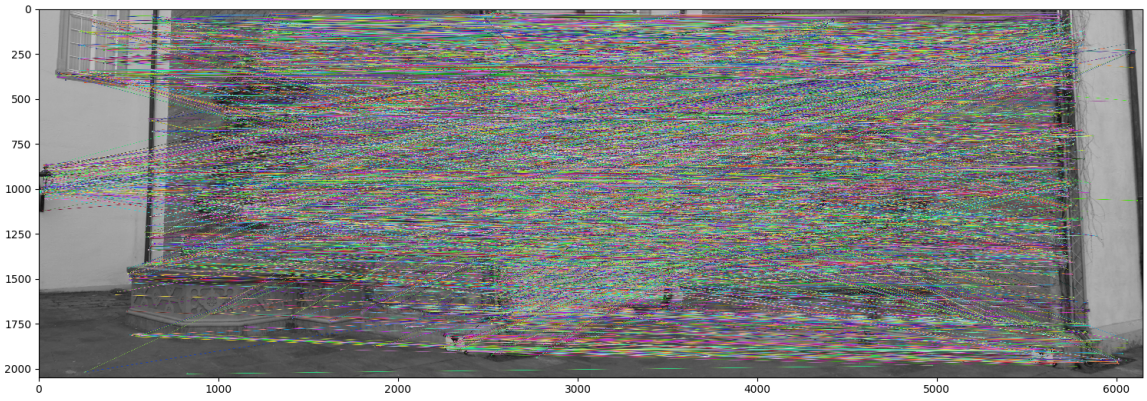

Figure 1: Matched feature points.

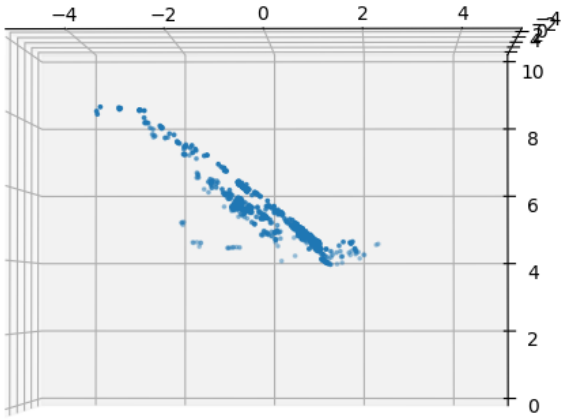

Figure 2: Reconstructed 3D points.

You will see that the result is not perfect; in a typical SfM pipeline, one would now run a bundle adjustment that would globally optimize the feature and camera positions.

Hand in:

Write a short report explaining the main steps of your implementation. The report should contain figures showing:

- **a)** Epipolar geometry of the image pair used for initialization
- **b)** Inlier and outlier matches for every image used in each step
- **c)** 3D image of the scene showing the triangulated points obtained using all 5 images.

Send the report together with your source code (but not the data) to `ajad.chhatkuli, paudel@vision.ee.ethz.ch`.