

CMSC733 Project 3: "Buildings Built in Minutes" (Structure from Motion)

Jack Rasiel
School of Computer Science
University of Maryland, College Park
Email: jrasiel@cs.umd.edu

I. METHODS

We give sample output for several steps of our pipeline:

A. Refining point matches via RANSAC

Problem encountered: tuning the threshold for the fundamental matrix.

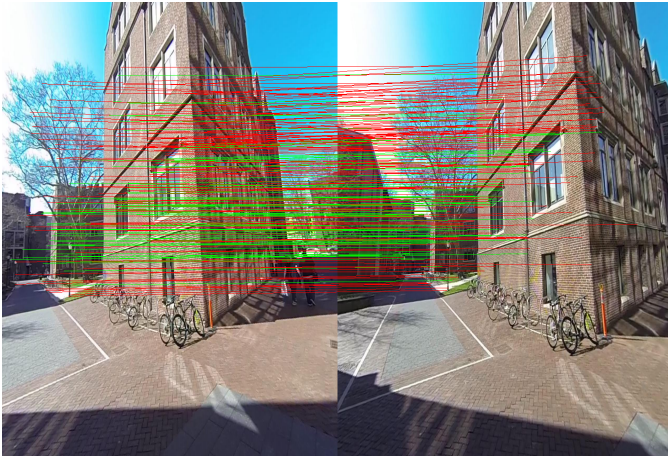


Fig. 1. Outlier rejection via RANSAC. Outliers are in red; inliers in green.

B. Current Status

Unfortunately, my code doesn't currently work. I implore the grader to look at my code, as I've implemented **every part of the pipeline very thoroughly**, with extensive commenting demonstrating full understanding of the project. In the end, I simply ran out of time debugging, and need to move on to the midterm. The code I've submitted won't run; an earlier version of my code, which doesn't include PnP (it does epipolar geometry-based registration for every pair of frames), *does* run, but doesn't produce good results. Example results are shown in fig2 and fig3.

C. Other observations

Debugging this project was quite challenging. We found the following practices and techniques useful:

- **Controlled test data.** We created a set of manual point correspondences on easily-identifiable points in the image. Specifically, for the first three images, we used the corners of windows on the building as correspondence



Fig. 2. Example frame from the debug dataset.

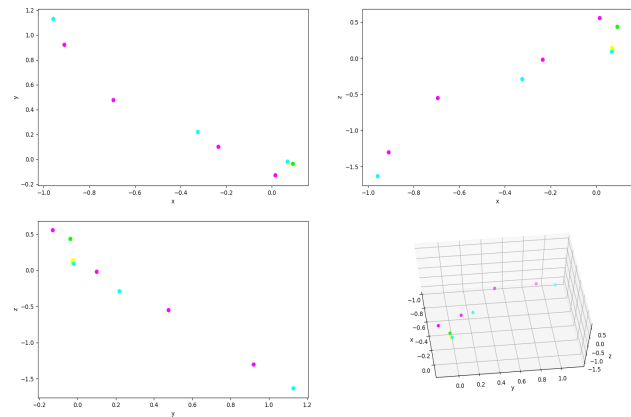


Fig. 3. Reconstruction results.

points (being sure to use two different faces of the building, so the points weren't all co-planar). By modifying this data in a controlled way, we tested different aspects of the pipeline (RANSAC, triangulation, PnP, etc.). Additionally, using relatively small test datasets reduced the run-time of the program, allowing for faster iteration.

- **Visualization.** We created visualizations for various stages of the pipeline, to help more easily identify at

which stages problems were starting. For example, using the test-data described above, we color-coded the points for each window in the final point cloud. This gave quick intuition as to the quality of our results.

II. VISUALSfM REFERENCE OUTPUT

For reference, here is the reconstruction of the same data, as given by VisualSfM:

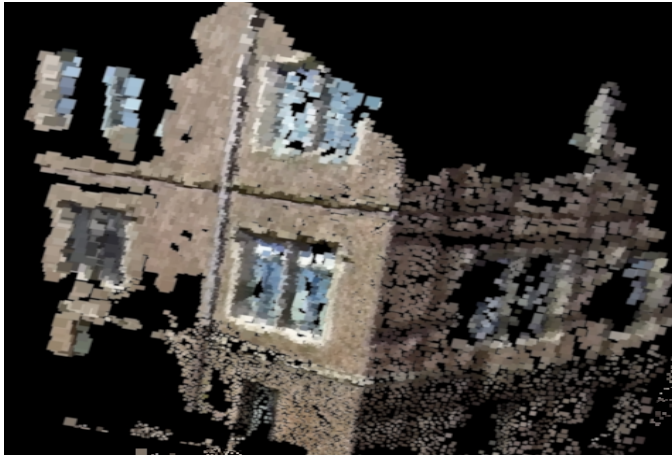


Fig. 4. VisualSfM Output