VISUAL ODOMETRY: MAPPING OUT THE CAMERA PATH

CS 585 CLASS CHALLENGE

Ariel Lee

April 2022

1 Introduction

The goal of this class challenge is to accurately estimate the path of a camera by recovering relative motion between successive frames. The camera's intrinsic parameters are provided along with preprocessed frames and ground truth coordinates. The only OpenCV functions used are for feature detection and reading images as numpy arrays. Equations not discussed in lecture are explicitly defined in this report.

2 Pipeline

All video frames are processed in a loop, starting with the first and second images.

2.1 Feature detection

Binary Robust Invariant Scalable Keypoints (BRISK) is used to detect feature points and descriptors between consecutive frames. K-Nearest Neighbor (kNN) matching is then utilized to extract the two best matches for each descriptor so that Lowe's ratio test can be applied [1]. Matches with the smallest distances are kept [3].

2.2 Fundamental matrix estimation using Random Sample Consensus (RANSAC)

After finding corresponding image points, the fundamental matrix is calculated and refined using the normalized eight-point algorithm and RANSAC. Eight random matches are selected and normalized by centering and scaling so that the mean of distances from the origin to the points equals $\sqrt{2}$. Then, the homogeneous least squares equation is solved by Singular Value Decomposition (SVD). The smallest singular value is removed to enforce the rank-2 constraint, and the fundamental matrix is transformed back to its original units. The number of inliers corresponding to the estimated fundamental matrix is determined through epipolar geometry and the Sampson distance, defined as:

$$\sum \frac{(x'^T F x)^2}{(Fx)_1^2 + (Fx)_2^2 + (F^T x')_1^2 + (F^T x')_2^2}$$

F = fundamental matrix, *x* = matching point in the first image, and *x'* = matching point in the second image. For each point *x* in the first image there exists a corresponding epipolar line *l'* in the second image. Any point *x'* that matches point *x* must lie on the epipolar line *l'* and satisfy the epipolar constraint. The Sampson distance is a first-order approximation of geometric error between a point's epipolar line and its corresponding point [1]. If the error is below a specified threshold, 0.92 for this model, then the point is considered an inlier. After 150 iterations, the fundamental matrix with the most inliers is chosen. Only the keypoints classified as inliers are used moving forward.

2.3 Essential matrix & camera pose estimation

The essential matrix is calculated from the fundamental and camera calibration matrices. To recover the camera position, SVD is performed on the essential matrix. The singular vector associated with the smallest singular value gives us the translation vector, or pose. However, we need to generate four possible position solutions since *U* and *V* are not guaranteed to be rotations—flipping both signs can still result in a valid SVD. Thus, both possible rotation matrices are paired with both possible signs of the translation vector. Additionally, if the determinant of the rotation matrix is -1, the signs of the rotation matrix and translation vector are flipped. To determine which of the four translation vector and rotation matrix pairs are correct, linear triangulation is performed on each to recover the possible 3D point clouds for the current frame. Individual 3D world coordinates are determined by corresponding feature points in both frames. The pose with the largest cheirality—number of points in front of the camera—is selected [3]. A 3D point *X* is in front of the camera if $r_3 \times (X - t) > 0$, since the distance must be positive, where r_3 is the *z*-axis rotation vector and *X = (x, y, z, 1)*. The final translation vector is multiplied by the given ground truth scale value. The pose matrix is then created by stacking the rotation matrix and translation vector column-wise and adding [0, 0,

0, 1] as the fourth row. To find the final coordinates of the camera for the current frame, the dot product of the current pose matrix and the previous pose matrix is calculated. The first three values in the last column represent the *x, y,* and *z* coordinates of the camera for the current frame.

3 Results & Discussion

A Mean Squared Error of 4.81 was achieved on the training set. This translated to a final score of 6.19 on gradescope. When calculating cheirality, including the condition that the z-coordinate of a 3D point must itself be positive—in addition to the constraint mentioned above—increased accuracy. Removing outliers by using the Sampson distance also increased accuracy. Oriented FAST and Rotated BRIEF (ORB) resulted in poor image matches, so BRISK was chosen because it is more robust to scale and rotation changes [2]. Implementing Lowe's ratio test after applying kNN matching increased accuracy over the initial brute force method [3]. Interestingly, most of the overall error comes from the y-coordinate, while the x and z-coordinates are estimated with higher precision. A plot of the final camera pose predictions and the ground truth coordinates is shown below in Figure 1.

A direct least squares solution to the Perspective-n-Point pose estimation was attempted using reprojection error as the loss. However, the computation time exceeded the 20 minute limit. Additionally, a previous submission using OpenCV to find the essential matrix resulted in a score of 115.79 on gradescope.

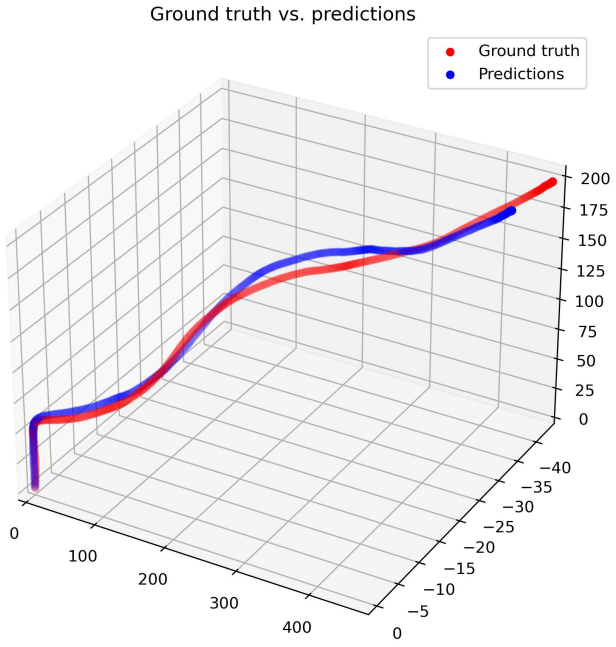

Figure 1: Camera path

4 References

1. Hartley, R., & Zisserman, A. (2004). Multiple View Geometry in Computer Vision (2nd ed.). Cambridge: Cambridge University Press. doi:10.1017/CBO9780511811685
2. S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2018, pp. 1-10, doi: 10.1109/ICOMET.2018.8346440.
3. Szeliski, R. (2021). Computer Vision: Algorithms and Applications (2nd. ed.). Springer-Verlag, Berlin, Heidelberg.