ESE650 Learning in Robotics, 2018 Spring
Final Project: Visual Odometry
Wudao Ling

Introduction

For ESE650 final project, I worked on Visual Odometry on KITTI dataset. Using a sequence of greyscale images, I need to estimate camera motion. I implemented and compared 2 different VO methods: Monocular 2D-2D VO and Stereo 2D-3D VO. Both of them are feature-based.

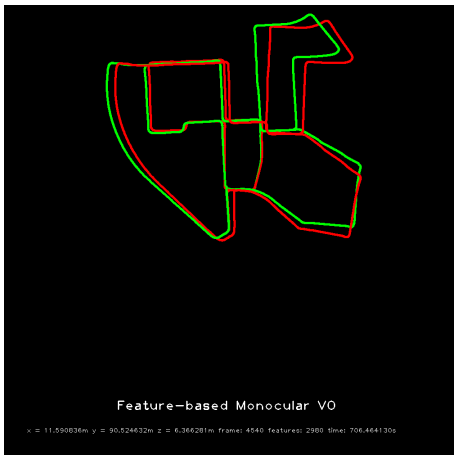
1 Dataset

KITTI (`http://www.cvlibs.net/datasets/kitti/eval_odometry.php`) is a well-known vision bench in autonomous driving area. KITTI's odometry data is collected by stereo cameras mounted on car, and it comes with callibration files, GPS ground truth as well as lidar data for hybrid methods.
In my project, I used greyscale data in consideration of speed and memory. And I didn't preprocess data.

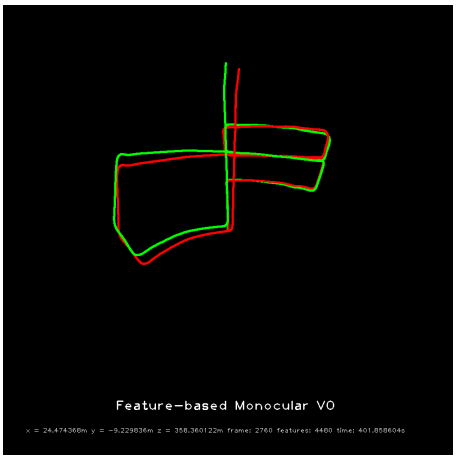
Figure 1: A example of KITTI greyscale odometry data

2 Monocular 2D-2D VO

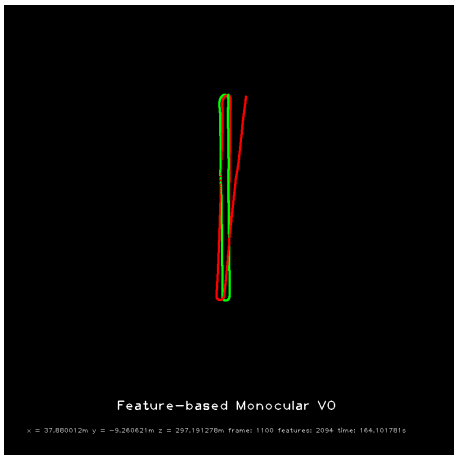
Monocular method only use single camera's data, generally it can be used to solve the motion angle but not motion scale in VO. In practice, monocular methods is always used in hybrid way with other sensor data such as IMU or lidar to overcome this problem.
In this part I use left camera for VO. The basic idea is using feature matching and epipolar geometry for motion estimation, the whole process never project 2D features to 3D landmarks. The scale of transformation is directly extracted from ground truth.



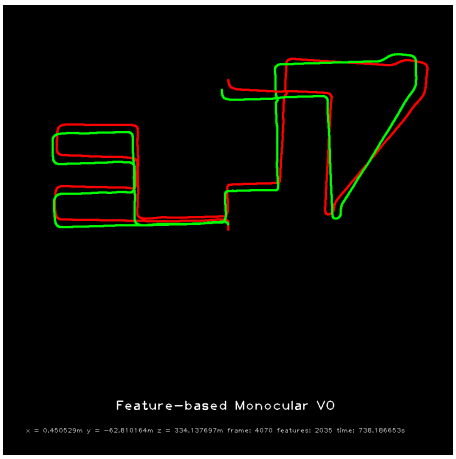
(a) sequence 0



(b) sequence 5



(c) sequence 6



(d) sequence 8

Figure 2: 2D-2D Mono VO

The pipeline is:

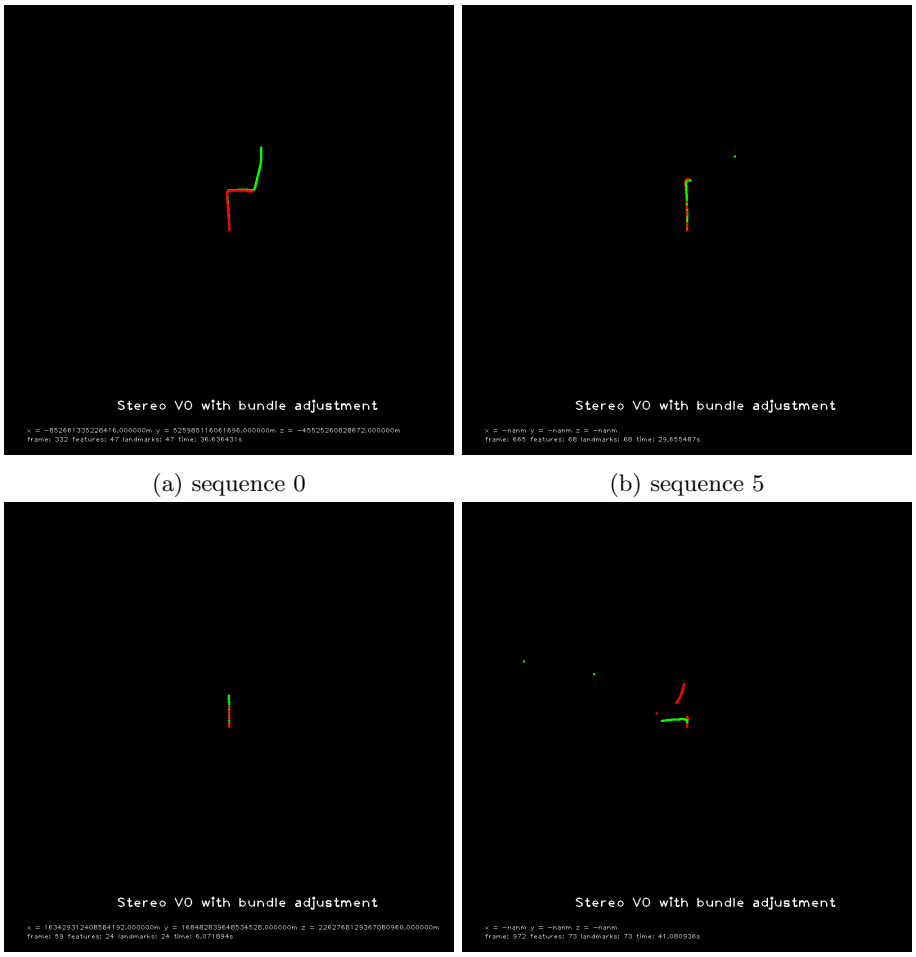
1. load calibration, ground truth and scales, init first image as keyframe
2. use FAST to extract features in keyframe
3. use KLT optical flow to track features in next frame, drop features that lose tracking
4. apply feature matchings to epipolar geometry, compute essential matrix and decompose to relative transformation between 2 frames
5. use relative transformation and last pose to update camera pose
6. if features drop under a threshold (2000), nominate current frame as a new keyframe and iterate from step 2, otherwise iterate from step 3

There are 11 sequences of image data with ground truth. I ran this method on all of them and get good results. As you can see, Figure 2 are several hardest sequences. (red trajectory)
This VO can serve as motion model of SLAM, because it's really lightweight and fast, and its results are accurate enough as an initial estimate. Then another measurement model could be used to refine the estimation.
However, this method doesn't fit if I need a visual SLAM solution. Because VO tends to gain drift error frame-to-frame over time, usually we handle this problem with Bundle Adjustment. Either offline or online, bundle adjustment need to record camera poses and 3D landmarks, then apply optimization to reduce reprojection error. 2D-2D method doesn't generate 3D landmarks, therefore I looked into next method.

3 Stereo 2D-3D VO

Stereo method use epipolar cameras' data. In KITTI dataset, left and right camera are already rectified, thus their orientation are always the same and their positions have a constant difference in X axis.
The pipeline is:

1. load calibration and ground truth
2. extract features and triangulate landmarks from key frame
    - use ORB to detect keypoints and compute descriptors in both left and right frame
    - use FLANN matcher to match them, do 0.7 ratio test on matches
    - with 2 camera projection matrix, triangulate matches to landmarks, transform landmarks from camera frame to world frame
3. track features and landmarks between previous left frame and current left frame
    - use KLT optical flow to track features, drop features and landmarks that lose tracking
4. given features in current frame and world landmarks, solve PnP problem with RANSAC to get absolute transformation from camera to world (camera pose)
5. record camera pose and landmarks, for a certain number of frame, do online bundle adjustment to optimize
6. iterate from step 2, in this implementation every frame is keyframe except number of features extracted is low or number of PnPRANSAC inliers is low.



(a) sequence 0

(b) sequence 5

(c) sequence 6

(d) sequence 8

Figure 3: 2D-3D Stereo VO

I was not able to achieve Bundle Adjustment due to time limit. This 2D-3D method without BA crashes very easily. I figured out that PnPRANSAC would easily output 0 inlier over time and give a weird transformation, this might result from stricter feature extraction. Instead of thousands of features in monocular method, ORB usually provide tens of features. But changing feature extraction may introduce another problem, too many features will lead to too many landmarks for BA, and a bunch of them could be duplicate.
Even though I failed to achieve a good performance with 2D-3D stereo VO this time. It will be a good starting point, I will look into fine-tuning to avoid crash, as well as bundle adjustment and loop closure.

References

[1] Scaramuzza, D. Tutorial on Visual Odometry *ETH Zurich Aerial and Service Robotics Summer School*, (2012), 1-94.

[2] McGarey, P. Visual Odometry (VO) *UofT CSC2541*, (2016), 1-36.