Computer Vision CSL7360 - Major Project Report

Visual Odometry

Github Repo [https://github.com/ayushabrol13/Visual-Odometry]

Name	Roll No.
Devyani Gorkar	*B20ME027*
Anjali Agarwal	*B20AI051*
Ayush Abrol	*B20AI052*
Aryan Tiwari	*B20AI056*

Introduction

Visual odometry is a method used in robotics and computer vision to estimate the motion of a camera or robot by analyzing images captured by its camera(s). It tracks key points or features in consecutive images and by analyzing how these features move over time, visual odometry algorithms can infer translation (movement along the x, y, and z axes) and rotation (changes in orientation) of the camera . Visual odometry can be carried out using both **monocular**(without absolute scale) as well as **stereo** cameras(absolute scale can be determined).

Problem Statement

Given a sequence of n consecutive images captured by a camera system$(I_1, I_2, ...I_n)$, where each image I_i represents the 2D projection of a 3D scene onto the image plane, and assuming known camera intrinsic parameters K and extrinsic parameters, the goal of visual odometry is to estimate the camera's trajectory $(T_1, T_2, ...T_n)$ with respect to global frame over the sequence, where T_k is (3×4) matrix given as

$$T_k = \begin{bmatrix} R_k & t_k^T \end{bmatrix}$$

The visual odometery problem can be solved using 2D-2D correspondances using epipolar geometery and 3D-2D correspondance using PnPRANSAC.

For monocular visual odometery, given a set of matching normalized image coordinates (p, p') the epipolar constraint equation is given as

$$p^T.E.p' = 0$$

The above equation is solved using five or 8 point algorithm along with RANSAC to obtain the accurate essential matrix E. Rotation and translation matrix is obtained using SVD of E.

$$E = [t]_x R$$

T_{mat} is the homogenous transformation matrix the represents camera motion between (i-1)th and ith frame.

$$T_{mat} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix}$$

T_{tot} is the homogenous transformation matrix representing the camera motion of the current frame with respect to the initial frame (global coordinate frame). T_{tot} is stored to record the trajectory of the camera.

$$T_{tot} = T_{tot}.T_{mat}^{-1}$$

For stereo visual odometery, given a sequence of n consecutive left and right images captured by a camera system$(I_{1l}, I_{2l}, ...I_{nl})$, $(I_{1r}, I_{2r}, ...I_{nr})$ seperated by a baseline b, where each image I_i represents the 2D projection of a 3D scene onto the image plane, and assuming known camera intrinsic parameters K and extrinsic parameters, the goal of visual odometry is to estimate the camera's trajectory $(T_1, T_2, ...T_n)$ with respect to global frame over the sequence.

The set of left and right images are used to compute the depth map of each pair.

$$Z = f * b/disp$$

Given a set of corresponding keypoints matches in left ith and (i+1)frame, objective is to find the 3D coordinates X_w of keypoints from the depth map and image coordinates (u, v) of ith frame.

$$\begin{gathered} X = Z * (u - c_x)/f_x \\ Y = Z * (v - c_y)/f_y \\ Z = Z \end{gathered}$$

Points expressed in the world frame X_w are projected into the image plane [u,v] using the perspective projection model Π and the camera intrinsic parameters matrix K.

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

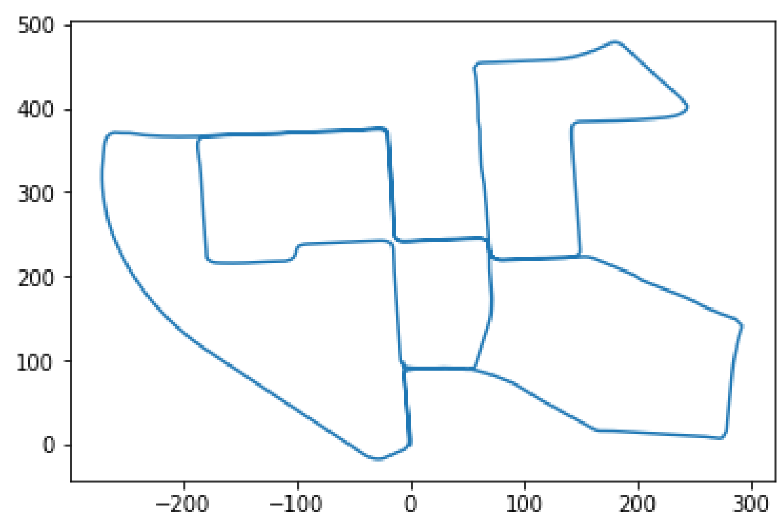
The PnP algorithm solves the above perspective projection equation using least squares method to obtain R and t. The camera's relative relative pose is accumulated to obtain the pose with resppect to initial frame.

Dataset

Dataset has images taken from left camera and right camera separated with a baseline b. Along with a series of image frames, intrinsic parameters, projection matrix and ground trajectory of the vehicle is also proivided.

> https://prod-files-secure.s3.us-west-2.amazonaws.com/dc21963f-00b6-45c8-b70f-ae2431dfddec/4db6f01f-03ab-4f6a-b8d5-1dc595b44b17/WhatsApp_Video_2024-04-27_at_4.30.22_PM.mp4

The ground truth.csv file has flattened transformation matrix $\begin{bmatrix} R & t \end{bmatrix}$ representing every camera frame pose with respect to intial camera frame. Below plot represents the actual trajectory of the camera throught the image frames.



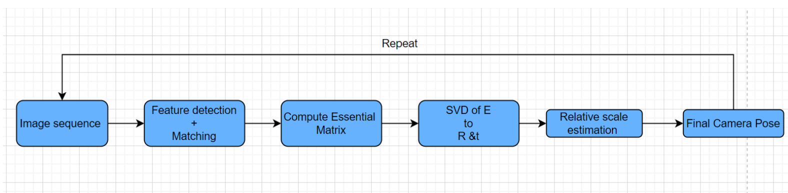
Ground truth Trajectory

Methodology :

In visual odometry, the initial frame serves as the reference or origin of the world coordinate system. As the camera moves through the scene, the relative motion between consecutive frames is estimated using visual odometry methods. This relative motion includes both rotation and translation between two consecutive frames. By accumulating these rotations and translations over time, we can compute the camera pose of the current frame (ith frame) with respect to the initial frame, which acts as the world coordinate system.

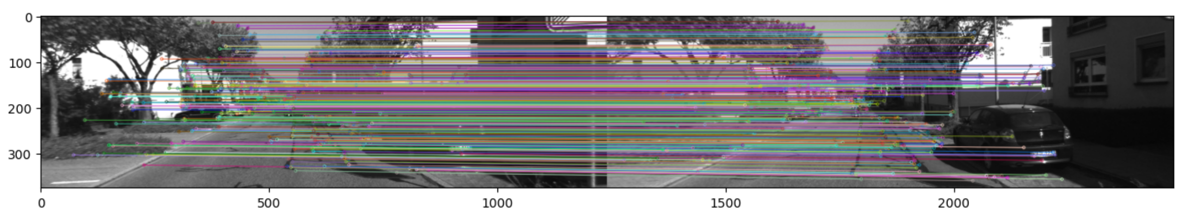
Visual Odometery can be computed using three main methods:

1. Monocular Visual Odometery using Feature matching:



In monocular visual odometery features matching is performed on consecutive left image frames.

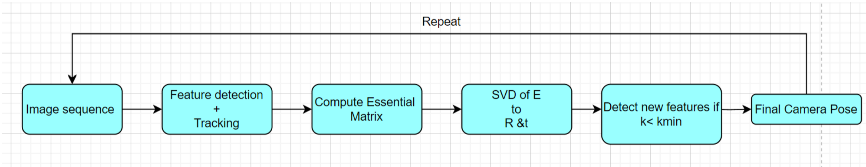
1. Intial frame homogeneous transformation matrix initialised as Identity matrix
2. Keypoints and descriptors were extracted from the (i)th and (i+1)th frame using a feature detection and description algorithm such as SIFT or ORB. FLANN based knn matcher used to match the keypoints in the consecutive frames



Feature Matching

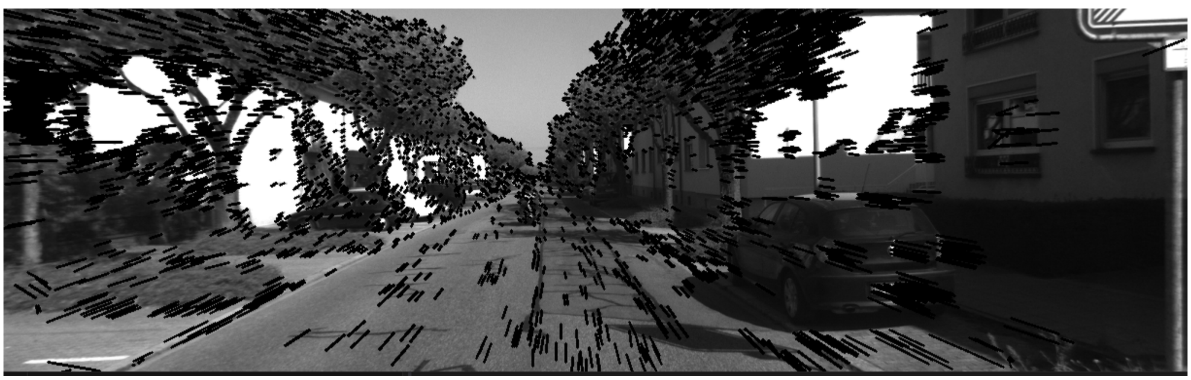
3. Essential matrix is computed using the intrinsic paramters of the camera and matched keypoints in the ith (i+1)th frame. Essential matrix is decomposed using SVD which results into R1, R2, t due to θ ambuiguity encountered in SVD.
4. The problem with monocular visual odometery is the unknown scale factor. Different pairs or rotation and translation matrices are used to obtain 3D landmarks using triangulation. Therefore we chose the $\begin{bmatrix} R & t \end{bmatrix}$ pair that has maximum landmarks with positive z coordinates is used.
5. Relative scale is computed by comparing the distances between consecutive points in each set and taking their mean ratio.
6. The final camera pose of the ith frame with respect to world frame is accumulated based on the relative motion computed from the above steps.

Monocular Visual Odometery using feature tracking



Kanade Lucas Optical flow method tracks features in consecutive image frames.

1. The first frame is processed by initializing key points using the FAST feature detector.
2. In the next frames, the detected features k1 are tracked using the Optical flow method to obtain k2 in next frame.
3. The tracked features(k1, k2) in consecutive frames are used to compute the essential matrix, the essential matrix decomposition gives the relative camera pose between two image frames.
4. If the average change in keypoint location > 5, then the relative camera pose is accumulated to obtain the camera pose in the world coordinate frame.
5. Current frame keypoints become the previous frame keypoints. Steps from 2-4 are repeated.

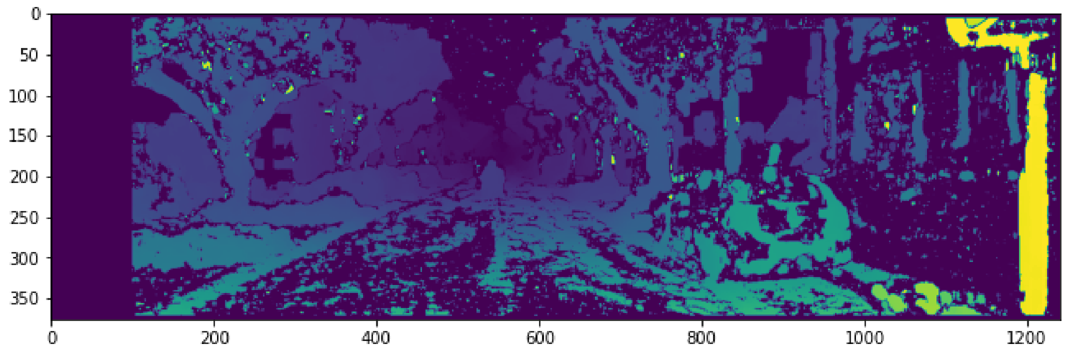


Feature tracking using optical flow

Stereo Visual Odometery

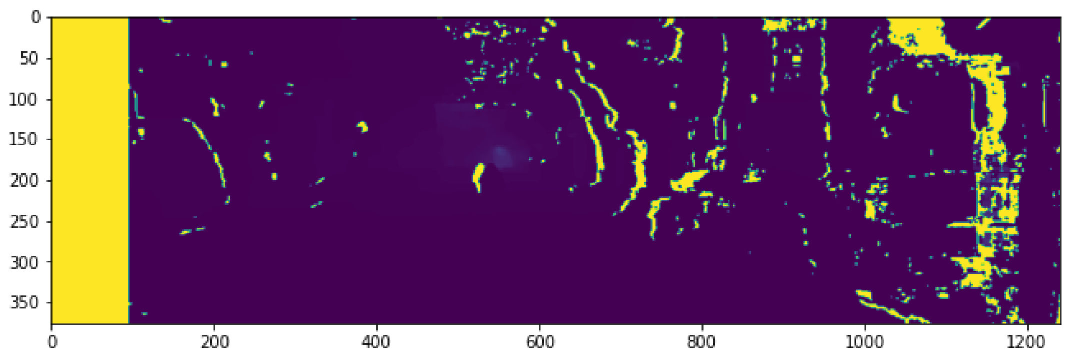
In stereo visual odometry, the left and right image frames are used to establish correspondence and compute the depth of images.

1. Detect features ith left image frame (i+1)th right image frame match the features using FLANN matcher. Filter the matches using Lowe’s ratio test to select only distinctively matching features.
2. Compute disparity map from ith left frame and ith right frame using Stereo block matching algorithm.



Disparity Map of the scene

3. Use the disparity map to compute depth map of the scene.



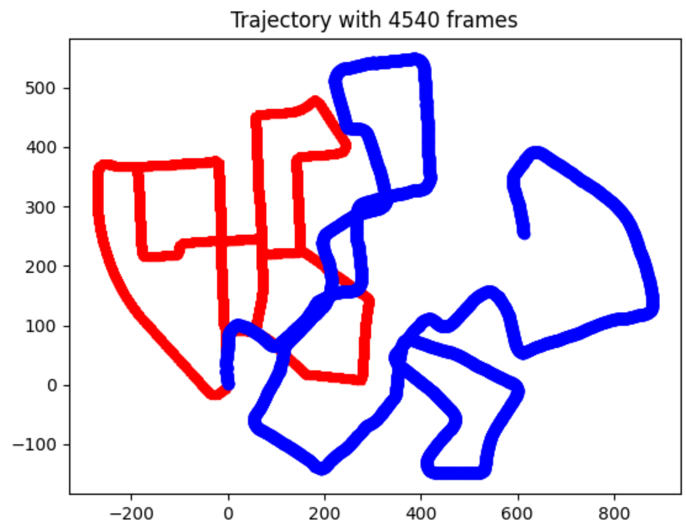
Depth map of scene

4. Use the depth map to obtain 3D coordinates of the key points detected in the image using the (u,v) image coordinates and intrinsic parameters. PnP RANSAC algorithm is used to obtain relative camera pose from the 3D-2D correspondence of the landmarks.
5. The relative camera pose is accumalated over previous camera pose to obtain pose in global coordinate frame.

Results:

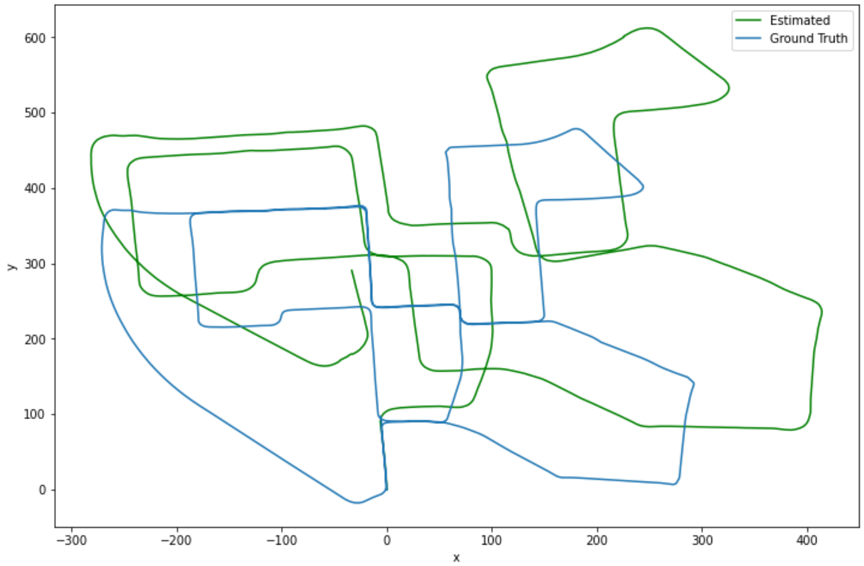
Technique	MSE Error
Mono VO using feature matching	254227.00112599428
MSE Error for Monocular VO using feature tracking	184299.7626976141
Stereo VO	53369.630670950115

1. Monocular Visual Odometery using Feature matching:

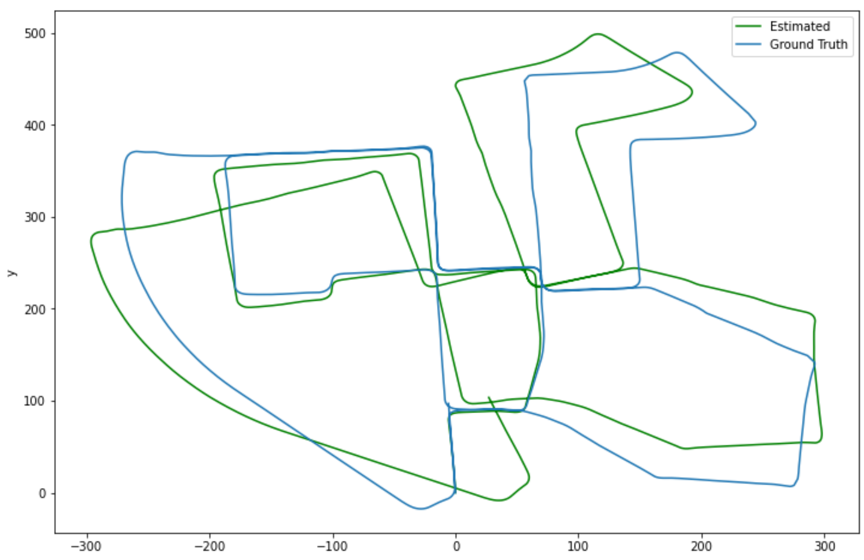


Red Trajectory- Ground Truth
Blue Trajectory - Visual odometery

2. Monocular Visual Odometery using Feature Tracking:



3. Stereo Visual Odometery:



Aspect	Monocular Odometry	Stereo Odometry
Depth Perception	Estimates depth indirectly, less accurate	Directly computes depth, more accurate
Scale Ambiguity	Suffers from scale ambiguity	Resolves scale ambiguity using stereo vision
Robustness	More susceptible to drift and errors	Generally more robust and accurate
Motion Estimation	Based on visual features and frame analysis	Uses stereo correspondences for motion estimation

Environment Compatibility	Suitable for general applications	Ideal for depth-rich and scale-aware scenarios
Image Usage	Utilizes a single set of images (mono)	Utilizes two synchronized images (stereo)
Computation	Less computationally intensive	More computationally intensive due to stereo matching and depth computation

Observations

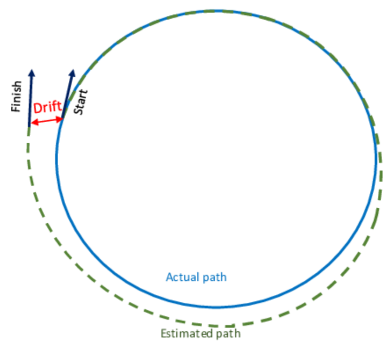
Stereo visual odometry outperforms monocular methods due to its ability to leverage depth information from stereo images, resulting in more accurate and robust trajectory estimation. However, it also requires more computational resources

1. **Stereo Visual Odometry:**
   - Provides the best results and is closest to the ground truth trajectory.
   - Utilizes depth information obtained from stereo images, resulting in more accurate 3D reconstructions.
   - Can handle occlusions and ambiguous feature matches better than monocular methods.
2. **Monocular Visual Odometry using Feature Tracking:**
   - Offers intermediate performance compared to stereo visual odometry and feature matching.
   - Tracks features across consecutive frames, providing a smoother trajectory estimation compared to feature matching.
   - Relies on the motion of tracked features to estimate camera motion, which can be affected by scene dynamics and feature quality.
3. **Monocular Visual Odometry using Feature Matching:**
   - Provides the least accurate results among the three methods.
   - Relies solely on matching features between frames, which can be prone to errors due to occlusions, lighting changes, and scene texture.
   - May struggle with maintaining feature correspondence over longer sequences, leading to drift in trajectory estimation.

Monocular visual odometry methods suffer from scale ambiguity, which means they cannot directly determine the absolute scale of the reconstructed scene. This ambiguity arises because monocular cameras capture images in 2D, lacking depth information.

All visual odometry methods are susceptible to drift accumulation, where small errors in pose estimation accumulate over time, leading to significant deviations from the ground truth trajectory. Drift can occur due to various factors, including:

- Noisy measurements or inaccuracies in feature detection, matching, or depth estimation.
- Inherent limitations of the estimation algorithms, such as assumptions about motion model or scene structure.
- Environmental factors such as lighting changes, occlusions, or dynamic scene elements



Conclusion:

In conclusion, visual odometry plays a crucial role in various applications such as robotics, augmented reality, and autonomous vehicles by enabling accurate localization and navigation based on visual cues. Throughout this discussion, we explored different approaches to visual odometry, including monocular methods using feature matching or tracking and stereo methods leveraging depth information from multiple viewpoints.

Each method has its strengths and limitations. Stereo visual odometry stands out for its ability to provide accurate 3D reconstructions and trajectory estimations, making it well-suited for applications requiring precise localization. Monocular visual odometry, while more computationally efficient, faces challenges such as scale ambiguity and drift accumulation. Feature tracking offers smoother trajectory estimation compared to feature matching but may struggle with dynamic scenes. Techniques such as sensor fusion, loop closure detection, and bundle adjustment are used to mitigate drift accumulation and improve overall performance.

References:

1. https://github.com/FoamoftheSea/KITTI_visual_odometry/blob/main/KITTI_visual_odometry.ipynb
2. https://cgarg92.github.io/Stereo-visual-odometry/
3. https://irvlab.cs.umn.edu/projects/dsvo-direct-stereo-visual-odometry