Assignment 1

Visual Odometry (VO) is a computer vision technique that addresses the problem of estimating the motion of a camera by analyzing the visual information obtained from its surroundings.

- Feature Tracking:
  - Visual Odometry often relies on the tracking of distinctive features in consecutive frames. These features can include corners, edges, or other high-contrast points.
  - Feature tracking involves identifying corresponding points between frames to establish correspondences over time.
- Camera Calibration:
  - Camera calibration involves determining the intrinsic parameters of the camera, such as focal length, principal point, and lens distortion.
  - Calibration ensures accurate mapping between the 3D world and 2D image coordinates.
- Epipolar Geometry:
  - Epipolar provides constraints on the possible locations of corresponding points in two images.
  - The epipolar geometry is exploited in Visual Odometry to establish correspondences and estimate the relative motion**.**
- Motion Estimation:
  - Motion estimation involves determining the camera's movement between frames. It can be achieved through techniques like optical flow, feature matching, or direct methods.

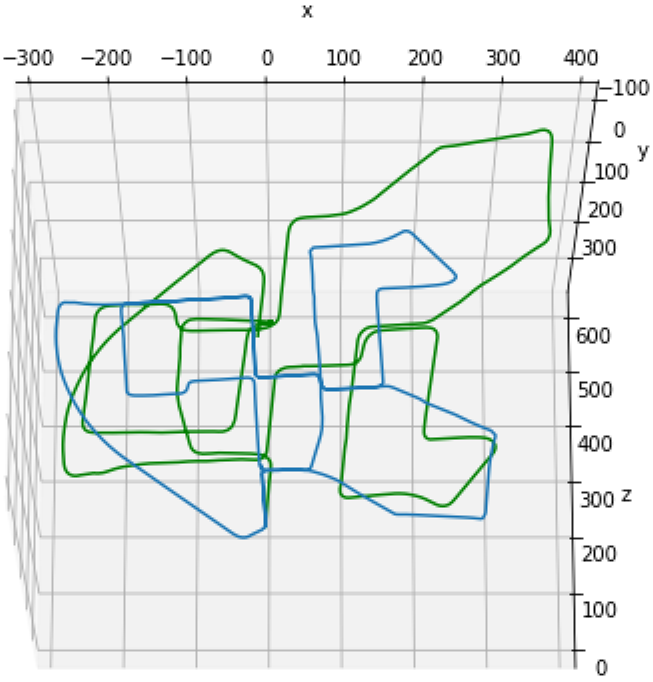
The dataset used for testing the visual odometry pipeline is KITTI dataset. It has 22 image sequences captured on different road conditions. Each sequence contains images from the left camera, right camera, calibration information, time information, and ground truth poses. The code is tested on a 00 image sequence having 4541 images. The ground truth poses are given in the form of a flattened 3x4 matrix transformation matrix between the global frame and the local frame. The global frame is the 0th image frame, and the local frame is the ith image.

In the code file, visual odometry is performed using three different approaches:

1. Monocular Visual odometry using feature matching between consecutive frames
2. Monocular Visual odometry using feature tracking in consecutive frames
3. Stereo visual odometry using feature matching between consecutive frames

The results and methodology for each case are explained below.

1. Monocular Visual odometry using feature matching between consecutive frames
    a. Features are detected using SIFT detector (ORB or SURF). The detector returns key points and corresponding descriptors of the image. These descriptors are further used to match features in two different images.
    b. The match_features function is defined to match features of images based on their descriptors. Two matchers can be used to test performance, Brute force and FLANN matcher.
    c. Lowe's ratio test is used to retain only strongly matched features between consecutive images. This test filters the features based on their ratio of distances. The matcher for each feature in the first image finds k features in the next images.
    The ratio of the distances between the descriptors of matched feature show how unique a particular feature is. If the ratio is less than a certain specified threshold, then the feature can be considered unique. If the ratio is close to 1, then the feature is not distinct and is discarded.
    d. The motion is estimated using Essential Matrix computation. The essential matrix, key points, and intrinsic parameters are used to recover the Rotation matrix and translation vector. This transformation is between the current frame and the previous frame. To obtain with respect to the global frame, the transformation is accumulated over consecutive frames.
    e. The following graph shows the ground truth trajectory and the estimated trajectory obtained from the above approach. The blue trajectory is the ground truth, and green is the estimated trajectory.
    f. To obtain this trajectory feature detector SIFT is used, Brute force feature matcher is used.

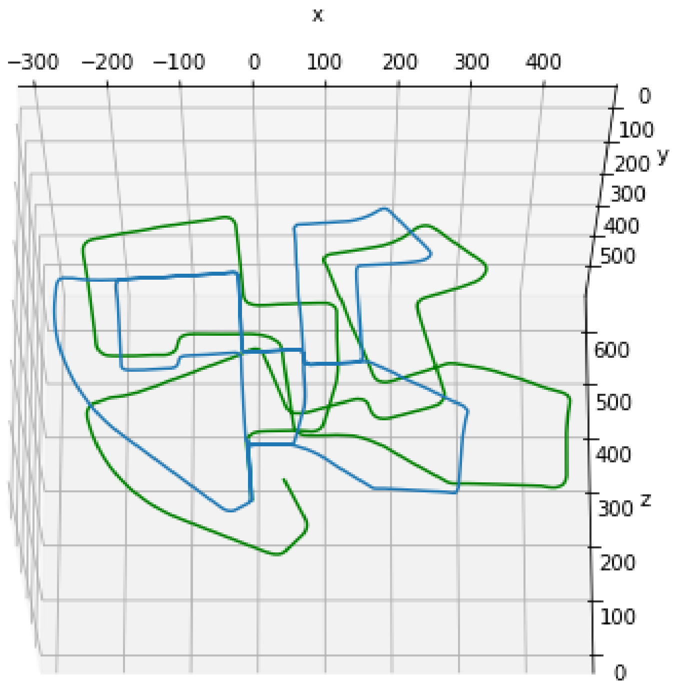


   g. The calculated mean squared error obtained through this approach is: 115364.029 m

2. Monocular Visual odometery using featuring tracking between consecutive frames:
   a. A function feature, tracking, is defined to track features between consecutive frames using Lucas-Kanade optical flow.
   b. The initial frames are processed to detect and track features. Fast feature detection is applied to the first frame. Essential Matrix, rotation, and translation are computed using RANSAC.
   c. The primary loop iterates over subsequent frames, systematically applying feature tracking and estimating the Essential Matrix, rotation, and translation. The condition for updating the camera pose is contingent on the mean feature displacement (
   change), introducing a dynamic mechanism for deciding when to account for changes in camera motion.
   d. The trajectory matrix (trajectory) is continually updated with the computed rotation and translation at each iteration, providing a comprehensive

record of the camera's poses over the course of the visual odometry process.

e. A conditional check is applied to maintain a sufficient number of features for accurate tracking. If the count of features drops below a specified threshold, Fast feature detection is employed to re-detect and integrate additional features into the tracking process. The cur_R and cur_t accumulate the transformation matrix over consecutive frames.
f. The transformation matrix (T_mat) is constructed from the rotation matrix (R) and translation vector (t). This matrix encapsulates the camera's transformation at each iteration, providing a comprehensive representation of its movement. Transformation matrix is stored in the trajectory matrix, storing all the poses.
g. The calculated mean squared error from this approach is: 220631.61 m



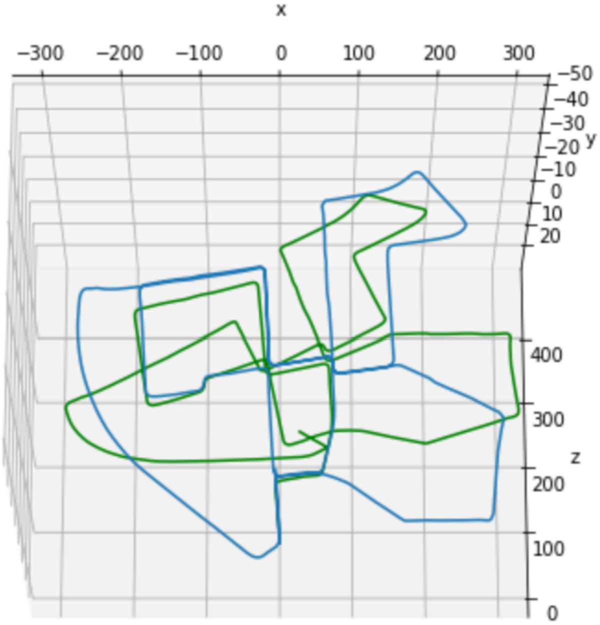
3. StereoVisual odometry:
   a. It uses current right, current left images and next left image to estimate the motion of the vehicle. Depth is computed from the left and right images. Feature matching is performed on current left and next left images.
   b. The depth image is computed from disparity, baseline and focal length of left camera. There are some pixels in the left camera image that are not present in the right camera image. To avoid dividing by zero or getting negative depth values, the disparity is set as 0.1. The computed depth map is used to obtain the 3D coordinates of the point in the world

c. The stereo_motion_est function estimates the motion based on the matched features of current and next images and the depth map through pinhole camera model. Image_points store the (u,v) coordinates of each matched feature. Depth of each matched feature is obtained from depth and 3D coordinates are computed from the intrinsic parameter.

d. From the obtained 3D points and its 2D coordinates in the next image open cv function PnPRansac computes the rotation matrix and transformation matrix between two frames. The rotation matrix is converted to desired format through Rodrigues function.

e. The final visual odometry pipeline computes for n frames and the results are accumulated in the T_tot matrix. The motion can be improved by tuning dist_threshold in while filtering the matches or by varying the detector type, disparity matcher.

   The calculated mean squared error from this approach is: 53544.896 m


x
−300
−200
−100
0
100
200
300
y
−50
−40
−30
−20
−10
0
10
20
z
400
300
200
100
0