Lab 4 Report - Stereo Camera-based Visual Odometry

Team E: Aditya Jain, Apurv Mishra, Praharsha Abbireddy

1 Introduction

In this lab, we estimate the pose of a moving vehicle using a stereo camera-based visual odometry. The major components of the lab include: use of inverse stereo model to convert matched keypoints into 3D points, outlier rejection using Random Sample Consensus (RANSAC) and, finally, pose estimation using scalar-weighted point cloud alignment.

2 Overall Architecture

The below sequential process is implemented for estimating the vehicle pose. For every image capture at current time *cur_image* at t and previous image *prev_image* at $t-1$:

1. Image de-wrapping and rectification of current left and right image frames, cur_image_left and cur_image_right
2. SIFT keypoint detection and matching in cur_image_left and cur_image_right
3. Matching keypoints from step 2. in $prev\_image\_left$ and $prev\_image\_right$ to get $matched\_features$ in *cur_image* and *prev_image*
4. $matched\_features$ is then converted to $3D\_points$ using inverse stereo model

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \frac{b}{u_l - u_r} \begin{bmatrix} 1/2(u_l + u_r) - c_u \\ \frac{f_u}{f_v}(1/2(v_l + v_r) - c_v) \\ f_u \end{bmatrix}$$

5. Use of RANSAC to filter out outliers in $3D\_points$
6. Finally, using scalar-weighted point cloud alignment to find transformation from the previous frame to the current frame to update vehicle pose from time $t-1$ to t.

$$T_{ba} = \begin{bmatrix} C_{ba} & -C_{ba} r_a^{ba} \\ 0^T & 1 \end{bmatrix}$$

7. This estimation is done frame-to-frame and the transformation matrices are recursively compounded for position at each time step to produce an estimate relative to the first frame.

   Note: The points at the previous time are defined in frame a and the points at the current time are defined in frame b.

   This report focuses on steps 5. and 6.

3 RANSAC

RANSAC is a general parameter estimation approach designed to cope with outliers in the input data. Iteratively, it generates candidate solutions by using the minimum number of data points (n) required to estimate the underlying model parameters and calculating the corresponding set of inliers from all the points for the estimated model given a tolerance $J_{threshold}$. The largest set of inliers is then used to calculate the final model parameters. For this lab, the below process is carried out for k iterations:

1. For the matched $3D\_points$ in frame t and $t-1$, n matched pairs are randomly sampled
2. C_{ba} and r_a^{ba} are calculated using scalar-weighted point cloud alignment (section 4)
3. Using C_{ba} and r_a^{ba}, classify every point i as an inlier or an outlier based on residual error in 1 and calculate the set of $inliers_k$

$$J^i = \frac{1}{2} w^i (p_b^i - C_{ba}(p_a^i - r_a^{ba}))^T (p_b^i - C_{ba}(p_a^i - r_a^{ba})) \leq J_{threshold} \tag{1}$$

4. Repeat this over for multiple trials and store the largest set of $inliers_k$.

Minimum three points are required to calculate T_{ba}, hence we take $n = 3$ and $w^i = 1$ is the weight assigned to point i. Theoretically, the number of required iterations

$$k = \frac{ln(1-p)}{ln(1-w^n)} \tag{2}$$

where p is the probability that atleast one sampled set has no outlier and w is the probability of a data point being an inlier. Quantitative analysis of the values of k and $J_{threshold}$ is done in the results section.

4 Pose Estimation using Point Cloud Alignment

For calculating the transformation that updates vehicle's pose from previous frame to the current frame, the following steps are executed:

1. Calculate the centroid of each point cloud and weights

$$p_a = \frac{1}{w} \sum_{j=1}^{J} w^j p_a^j$$

$$p_b = \frac{1}{w} \sum_{j=1}^{J} w^j p_b^j$$

$$w = \sum_{j=1}^{J} w^j$$

where w_j is the scalar weight for point j and equal to 1, and J is the total number of $matched\_features$ being considered in each point cloud.

2. Calculate the outer product or data matrix

$$W_{ba} = \frac{1}{w} \sum_{j=1}^{J} w^j (p_b^j - p_b)(p_a^j - p_a)^T$$

3. Apply Singular-Value Decomposition for satisfying the constraint, i.e. $detC = 1$, not -1

$$VSU^T = W_{ba} \text{ where } U^T U = V^T V = 1$$

4. The unknown rotation is defined as

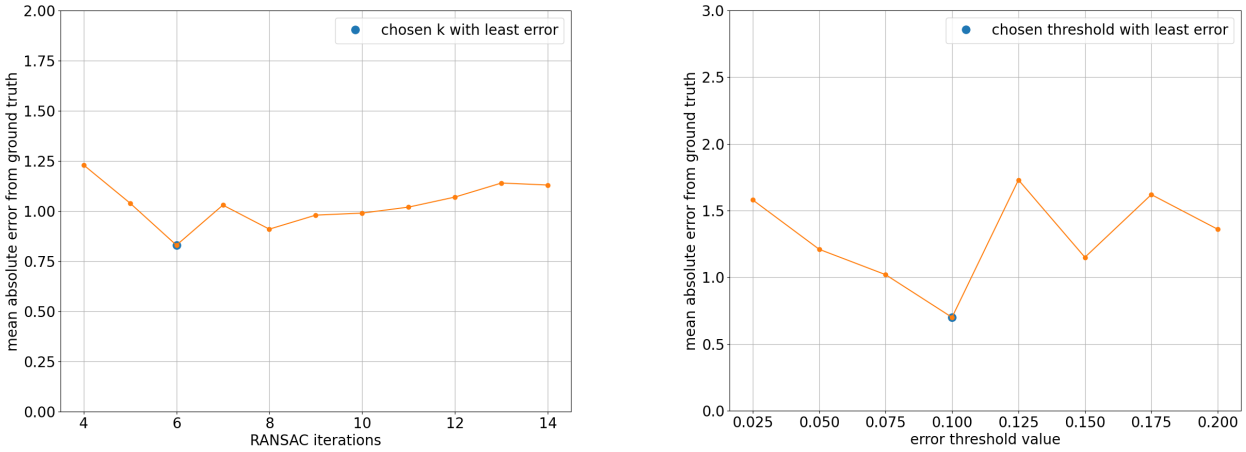
$$C_{ba} = V \begin{bmatrix} 1, & 0, & 0 \\ 0, & 1, & 0 \\ 0, & 0, & detU\,detV \end{bmatrix} U^T$$

and the unknown translation is defined as

$$r_a^{ba} = C_{ba}^T p_b + p_a$$

5 Results

We perform experiments to determine the required number of iterations k and the error threshold $J_{threshold}$, the results of which are shown in the plots in Figure 1. The metric chosen for evaluation is the mean absolute error or deviation from the ground truth. Thus, the parameters chosen correspond to the value which results in least absolute error. Note: the unit of error is same as the unit used for the provided ground truth.



(a) Iterations, k

(b) Error threshold, $J_{threshold}$

Figure 1: Plots for RANSAC parameters against the mean deviation of visual odometry from ground truth

The empirically chosen values for RANSAC are: $k = 6$ and $J_{threshold} = 0.1$. Note: the value k was also calculated from equation 2, taking $p = 0.9$ and $w = 0.7$, and gives the same result.

The final output is a plot between the ground truth and the trajectory calculated from visual odometry. Figure 2 shows the result and the drift in the estimated vehicle pose is evident as time progresses. The reason for this drift from the ground truth is discussed in the next section.

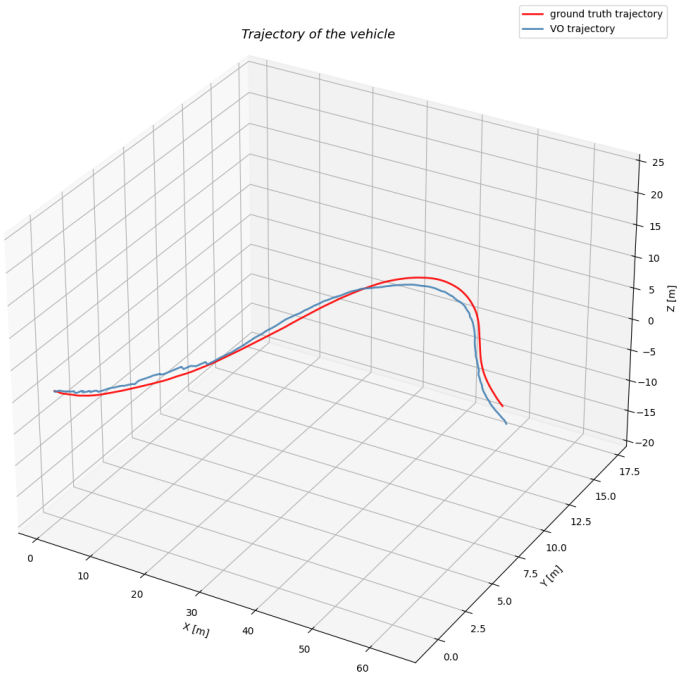


Figure 2: Vehicle trajectory comparison between ground truth and visual odometry

6 Discussion

Visual odometry is a *dead-reckoning* technique, hence, the error accumulates and the estimated pose drifts away significantly from the ground truth over a period of time. One possible way to address this issue without an external localization system is *loop closure*. If we return to a previously seen frame in the environment, we can correct for all the pose estimates that were calculated since the frame was last seen.