Visual Odometry using RGBD Camera

Sujoy Paul
University of California, Riverside, CA 92507
spaul003@ucr.edu

Abstract

RGB-D camera not only provides color images, but also provide depth of each pixel in the image, thus serving as a rich source of information for several robotics and computer vision tasks. This project develops a system to estimate the 6-DoF ego-motion of a camera using only RGBD images as input. A frame by frame approach is employed to detect the motion of the camera. The estimated motion in each pair of frames is "integrated" to obtain the camera pose in the global frame. Experimental analysis is done on a publicly available dataset.

1. Introduction

Visual odometry is one of the several ways of estimating the ego-motion of a camera, with images being the input. Visual odometry is advantageous over wheel odometry due to the wheel slippage problem of the former. Moreover, visual odometry can be used in estimating motions of drones, where wheel odometry is not possible. Low cost, light weight and rich amount of information (RGB and depth) provided by RGB-D camera has been the reason of attraction towards this type of sensors for visual odometry.

2. Overview of the Framework

A pictorial representation of the proposed framework is presented in Fig. 1. The assumptions of the proposed method is that the scene being observed is stationary w.r.t. the global frame and only the camera moves. At first the SURF features are extracted from two frames k and $k + 1$. Thereafter, they are matched to obtain obtain pairs of matched points between the two frames. The 3D co-ordinates of each feature points are extracted using the depth information.

After obtaining the 3D locations of each feature points, the homogeneous transformation matrix ${}^{C_k}T_{C_{k+1}}$ is obtained by using Gauss-Newton minimization coupled with RANSAC to reject outliers. The 6-DoF pose of the camera in the global frame is obtained by multiplying the individual transformation matrices. To start with, feature extraction and matching is discussed next.

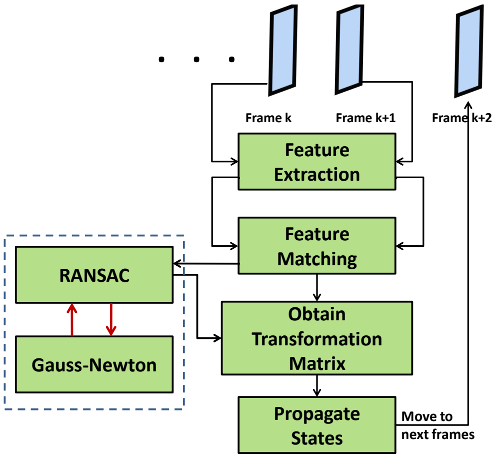


Figure 1: This is a pictorial representation of the flow of the proposed method.

3. Feature Extraction and Matching

The first step of the framework is feature extraction. Given two images at time step k and $k + 1$, the Speeded Up Robust Features(SURF) [1] are extracted from them. SURF feature extraction is claimed to be not only faster than SIFT features, but also robust against several image transformations. SURF uses a variant of the Hessian matrix based approach to detect the interest points. It also uses a scale-space representation in order to have several types of invariances in the detected features. It then uses a Haar wavelet based approach to extract the features. Finally, for each feature position a 64×1 vector is obtained as the feature descriptor.

Once the features are extracted from the two images, feature matching is done to obtain the correspondence between the feature points. Sum of Squared Differences (SSD) between the feature descriptors is used as a metric to match the features. A threshold is used as a first step of matching, i.e., if the minimum SSD is lesser than a threshold, only then that feature pair is considered for matching. The Nearest Neighbor Ratio test is used in the second step of matching. In this test, if the ratio of the smallest SSD and the second smallest SSD is lesser than a threshold, only then it is considered to be a valid match. These 3D co-ordinates of the matched features are obtained next.

4. 3D Co-ordinates

The feature matching procedure discussed above produces a set of pair of points having 2D pixel co-ordinates. The 3D co-ordinates of these points as observed from the camera frame can be obtained using the depth channel of RGB-D camera. Consider f_x, f_y, c_x and c_y to be the intrinsic parameters of the camera, then the (X, Y, Z) co-ordinates of the pixel co-ordinates (u, v) can be obtained as follows [1],

$$\begin{aligned} Z &= \frac{D(u,v)}{s} \\ X &= \frac{u - c_x}{f_x} Z \\ Y &= \frac{v - c_y}{f_y} Z \end{aligned} \tag{1}$$

where $D(u, v)$ is the depth image intensity values and s is a scaling factor from intensity to distance. Using the 3D co-ordinates of the matched features points, the motion is estimated as discussed next.

5. Motion Estimation

The goal is to obtain the 6-DoF camera pose in the global frame from the 3D co-ordinates of the matched features of the frames. At each time instant, the homogeneous transformation matrix ${}^{C_k}T_{C_{k+1}}$ mentioned below is obtained using image frame k and $k + 1$.

$${}^{C_k}\boldsymbol{T}_{C_{k+1}} = \begin{bmatrix} {}^{C_k}\boldsymbol{R}_{C_{k+1}} & {}^{C_k}\boldsymbol{t}_{C_{k+1}} \\ 0_{1\times 3} & 1 \end{bmatrix} \tag{2}$$

${}^{C_k}\boldsymbol{R}_{C_{k+1}}$ and ${}^{C_k}\boldsymbol{t}_{C_{k+1}}$ are the rotation matrix and translation vector from frame C_{k+1} to C_k. The translation vector contains ${}^{C_k}\boldsymbol{t}_{C_{k+1}} = [{}^{C_k}t_{xC_{k+1}} \quad {}^{C_k}t_{yC_{k+1}} \quad {}^{C_k}t_{zC_{k+1}}]^T$. Euler angle representation have been used for the rotation matrix as follows,

$${}^{C_k}\boldsymbol{R}_{C_{k+1}} = \boldsymbol{R}_z({}^{C_k}\alpha_{C_{k+1}})\boldsymbol{R}_y({}^{C_k}\beta_{C_{k+1}})\boldsymbol{R}_x({}^{C_k}\gamma_{C_{k+1}}) \tag{3}$$

where $\boldsymbol{R}_{(a)}(b)$ represent the rotation matrix around axis a by amount b.

Using the homogeneous transformation matrices at each time step, the camera pose of time step $k + 1$ in the global frame can be expressed as follows,

$${}^{G}\boldsymbol{T}_{C_{k+1}} = {}^{G}\boldsymbol{T}_{C_k}{}^{C_k}\boldsymbol{T}_{C_{k+1}} \tag{4}$$

We need to obtain the matrix ${}^{C_k}T_{C_{k+1}}$ at each time instant. A Maximum-Likelihood estimate of the same is obtained as discussed next.

5.1. Maximum Likelihood Estimate of Camera Pose

At each time step, given a set of correspondence points from two image frames, the goal is to obtain an estimate the following vector,

$$\boldsymbol{x} = [{}^{C_k}\alpha_{C_{k+1}} \quad {}^{C_k}\beta_{C_{k+1}} \quad {}^{C_k}\gamma_{C_{k+1}} \quad {}^{C_k}\boldsymbol{t}_{C_{k+1}}]^T \tag{5}$$

Consider $\{{}^{C_k}\boldsymbol{p}_i, {}^{C_{k+1}}\boldsymbol{p}_i\}_{i=1}^{N}$ to be the N pair of feature points observed in the two camera frames. They can be represented by an observation model as follows,

$${}^{C_k}\boldsymbol{p}_i = {}^{C_k}\boldsymbol{R}_{C_{k+1}}{}^{C_{k+1}}\boldsymbol{p}_i + {}^{C_k}\boldsymbol{t}_{C_{k+1}} + \boldsymbol{n}_i \tag{6}$$

where noise $\boldsymbol{n}_i \sim \mathcal{N}(0, \boldsymbol{Q}_i)$ and mutually independent. The maximum-likelihood (ML) estimate of $\boldsymbol{x}$ is obtained as follows,

$$\begin{aligned} \hat{\boldsymbol{x}}_{ML} &= \arg\max_{\boldsymbol{x}} \sum_{i=1}^{N} \log p({}^{C_k}\boldsymbol{p}_i; \boldsymbol{x}) \\ &= \arg\min_{\boldsymbol{x}} \sum_{i=1}^{N} \frac{1}{2}({}^{C_k}\boldsymbol{p}_i - h_i(\boldsymbol{x}))^T \boldsymbol{Q}_i^{-1}({}^{C_k}\boldsymbol{p}_i - h_i(\boldsymbol{x})) \end{aligned} \tag{7}$$

where the last step is assuming $p({}^{C_k}\boldsymbol{p}_i; \boldsymbol{x})$ to have a Gaussian distribution and $h_i(\boldsymbol{x}) = {}^{C_k}\boldsymbol{R}_{C_{k+1}}{}^{C_{k+1}}\boldsymbol{p}_i + {}^{C_k}\boldsymbol{t}_{C_{k+1}}$. The above minimization problem is solved using Gauss-Newton method.

The Gauss-Newton method is an iterative minimization procedure which starts with an initial guess of the estimate $\boldsymbol{x}^{(0)}$ and updates it in each iteration. The $k + 1^{th}$ iteration of Gauss-Newton can be represented as follows,

$$\begin{aligned} \boldsymbol{x}^{(k+1)} = \boldsymbol{x}^{(k)} + \Big(\sum_{i=1}^{N} \boldsymbol{H}_i^T \boldsymbol{Q}_i^{-1} \boldsymbol{H}_i\Big)^{-1} \\ \Big(\sum_{i=1}^{N} \boldsymbol{H}_i^T \boldsymbol{Q}_i^{-1}(\boldsymbol{z}_i - \boldsymbol{h}_i(\boldsymbol{x}^{\boldsymbol{k}})\Big) \end{aligned} \tag{8}$$

[1] http://vision.in.tum.de/data/datasets/rgbd-dataset/file_formats

where $\boldsymbol{H}_i = \frac{\partial \boldsymbol{h}_i}{\partial \boldsymbol{x}}|_{\boldsymbol{x}^{(k)}}$ and $\boldsymbol{z}_i = {}^{C_k}\boldsymbol{p}_i$. The matrix $\boldsymbol{H}_i$ can be expressed as follows [2],

$$\begin{aligned}
\boldsymbol{H}_i &= \begin{bmatrix} \boldsymbol{h}_{\boldsymbol{\alpha}} & \boldsymbol{h}_{\boldsymbol{\beta}} & \boldsymbol{h}_{\boldsymbol{\gamma}} & \boldsymbol{H}_{\boldsymbol{p}} \end{bmatrix} \\
\boldsymbol{h}_{\boldsymbol{\alpha}} &= \frac{\partial \boldsymbol{h}_i}{\partial^{C_k}\alpha_{C_{k+1}}} = \boldsymbol{R}_z \boldsymbol{J}_1 \boldsymbol{R}_y \boldsymbol{R}_x {}^{C_{k+1}}\boldsymbol{p}_i \\
\boldsymbol{h}_{\boldsymbol{\beta}} &= \frac{\partial \boldsymbol{h}_i}{\partial^{C_k}\beta_{C_{k+1}}} = \boldsymbol{R}_z \boldsymbol{R}_y \boldsymbol{J}_2 \boldsymbol{R}_x {}^{C_{k+1}}\boldsymbol{p}_i \\
\boldsymbol{h}_{\boldsymbol{\gamma}} &= \frac{\partial \boldsymbol{h}_i}{\partial^{C_k}\gamma_{C_{k+1}}} = \boldsymbol{R}_z \boldsymbol{R}_y \boldsymbol{R}_x \boldsymbol{J}_3 {}^{C_{k+1}}\boldsymbol{p}_i \\
\boldsymbol{H}_{\boldsymbol{p}} &= \frac{\partial \boldsymbol{h}_i}{\partial^{C_k}\boldsymbol{t}_{C_{k+1}}} = \boldsymbol{I}_3
\end{aligned} \tag{9}$$

where,

$$\boldsymbol{J}_1 = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \boldsymbol{J}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \boldsymbol{J}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

One problem of directly using all the matched feature points to obtain the estimate as discussed above is that there may be outliers in the matching, due to which the estimate may deviate considerably from the true value. For this reason, RANSAC is used to get rid of the outliers as discussed next.

5.2. RANSAC

Random Sample Consensus (RANSAC) [2] is a method of estimating the parameters of model with data containing outliers. First, it iterates over L iterations choosing a small subset of observations to obtain the parameters. This subset is randomly sampled from the entire set of observations, in each iteration. A criteria is defined to evaluate each randomly chosen subset. After all the iterations, the subset having the best values according to the defined criteria is selected and then the parameters are re-estimated using that subset of observations along with the observations which are near the subset and satisfy another defined criteria (considered to be inliers). The algorithm of RANSAC used in this work is described as follows. Consider the set of matched features $\mathcal{U} = \{{}^{C_k}\boldsymbol{p}_i, {}^{C_{k+1}}\boldsymbol{p}_i\}_{i=1}^{N}$.

```
for $i = 1 \ldots L$
    Step 1. $U$ = randomly obtain $n$ observations from $\mathcal{U}$
    Step 2. Obtain $\hat{\boldsymbol{x}}_{ML}$ (as discussed)
    Step 3. Store Median of Squared Residuals (MedS)
end
Step 4. Obtain subset $U^*$ having the Least MedS (LMedS)
Step 5. Perform a Mahalanobis Gating Test to obtain a set of observations $O$ nearby the observations in $U^*$, i.e., $\{{}^{C_k}\boldsymbol{p}_i, {}^{C_k}\boldsymbol{p}_i\} \in O$ if, $\boldsymbol{r}_i^T \boldsymbol{S}_i^{-1} \boldsymbol{r}_i \leq d$, where $\boldsymbol{r}_i$ is the residual of the $i^{th}$ observation using the estimated parameters in $U^*$, $\boldsymbol{S}_i = \boldsymbol{H}_i \boldsymbol{P}_{ML} \boldsymbol{H}_i^T + \boldsymbol{Q}_i$ and $d$ is a threshold obtained from the chi-squared distribution.
Step 6. Obtain $\hat{\boldsymbol{x}}_{ML}$ using the set of observations $O$.
```

Once the maximum likelihood estimate $\hat{\boldsymbol{x}}_{ML}$ is obtained, ${}^{C_k}\boldsymbol{R}_{C_{k+1}}$ is obtained using Eqn. 3, then ${}^{C_k}\boldsymbol{T}_{C_{k+1}}$ is obtained using Eqn. 2 and ${}^{G}\boldsymbol{R}_{C_{k+1}}$ is obtained using Eqn. 4. The results produced by the framework is presented next.

6. Experimental Results

Dataset. The publicly available RGBD SLAM dataset offered by TUM [3] [3] is used for experimental analysis. More specifically, the video named freiburg3_ long_ office_ householdhave is used. Details of the video are presented in Table 1. The intrinsic parameters of the camera required for obtaining the 3D co-ordinate of a pixel of the image is also provided in the dataset.

Table 1: Video details

Duration	87.09s
Ground-truth trajectory length	21.455m
Avg. translational velocity	0.249m/s
Avg. angular velocity	10.188deg/s
Trajectory dim.	5.12m x 4.89m x 0.54m

The ground truth provided in the dataset have the true global position of the camera as well as its orientation expressed in quaternions. Conversion from the euler angles to quaternions is done in order to perform comparison with the ground truth.

Details on RANSAC. The details of the parameters of RANSAC are presented in Table 2. Based on these, the no. of iterations L is computed as follows,

$$L = 1.5 * \frac{\log(1 - P_g)}{\log(1 - (1 - \epsilon)^m)} \tag{10}$$

Table 2: RANSAC parameters

% of outliers (ϵ)	0.3
No. of obs. per sample (m)	3
Prob. atleast 1 subset has all inliers (P_g)	0.99

The noise covariances $\boldsymbol{Q}_i = 0.01 * \boldsymbol{I}_3$ has been found on a trial-and-error basis. It may be noted that different combinations were tried on the diagonal (instead of all 0.01),

[2] Please note that all the derivatives are evaluated at $\boldsymbol{x}^{(k)}$.

[3] http://vision.in.tum.de/data/datasets/rgbd-dataset/download

and some combination gave better results (compared to the results presented in this report), for a particular set of DoF *only*, but not for all the DoFs. For this reason, for all the results presented hereafter, the above mentioned matrix have been used, which produced consistent results for all the DoFs. Also, it may be noted that the results may be better than what is presented here if the true noise covariance is known.

Comparisons. The obtained tracks along with the ground truth [4] are presented in Fig. 2, 3 and 4. Fig. 2 provides the 2D track and detailed tracks can be found in Fig. 3 and 4. The proposed framework is executed with various time intervals (δ) between the frames as indicated in the plots. For example, $\delta = 3$ means that the following pairs of frames have been considered for estimating the pose $1 - 4$, $4 - 7, \ldots$. As, only odometry is done in this work, the error builds up as we propoagate the estimates in each time step using Egn. 4. It may be noted that due to this reason, the error decreases as δ is increased. However, δ cannot be increased indefinitely as beyond a certain limit, the number of features matched between two frames will drop beyond the required number of matched features to properly obtain the rotation and translation estimates. It may be noted that the value of δ up to which the error decreases depends on the speed of the camera movement. As the speed of the camera increases, δ should be kept smaller to obtain a considerable number of matched features.

Comparison of the RANSAC based approach is also done with Iterative Closest Point algorithm (ICP) [4]. The results are presented in Fig. 2 and Fig. 5. ICP performs well, but a little worse compared to RANSAC (for the video used). Moreover, it may be observed that as δ increases, the performance of ICP degrades because the initial match becomes worse (as no odometry measurement is used for the initial match and direct mutual consistency matching has been used) and thus convergence to a value near the true value is not always guaranteed.

References

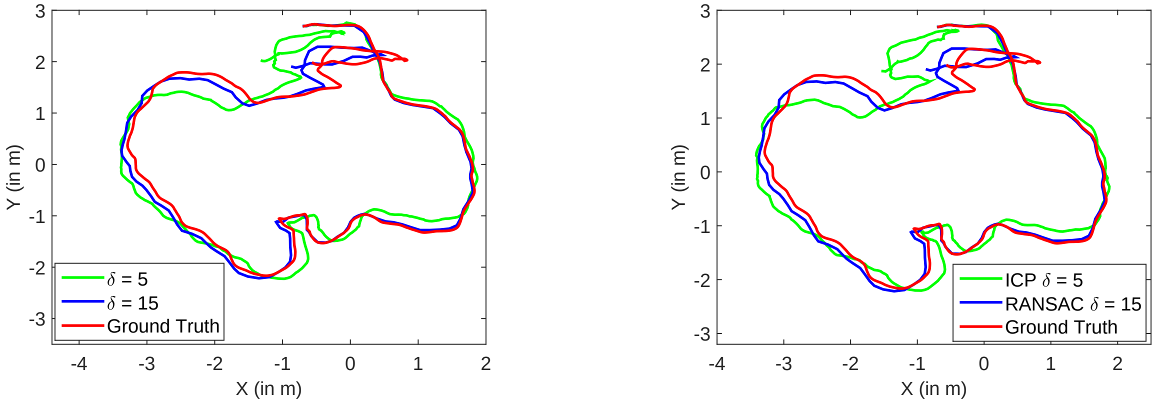
[1] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *Computer vision–ECCV 2006*, pages 404–417. Springer, 2006.

[2] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[3] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 573–580. IEEE, 2012.

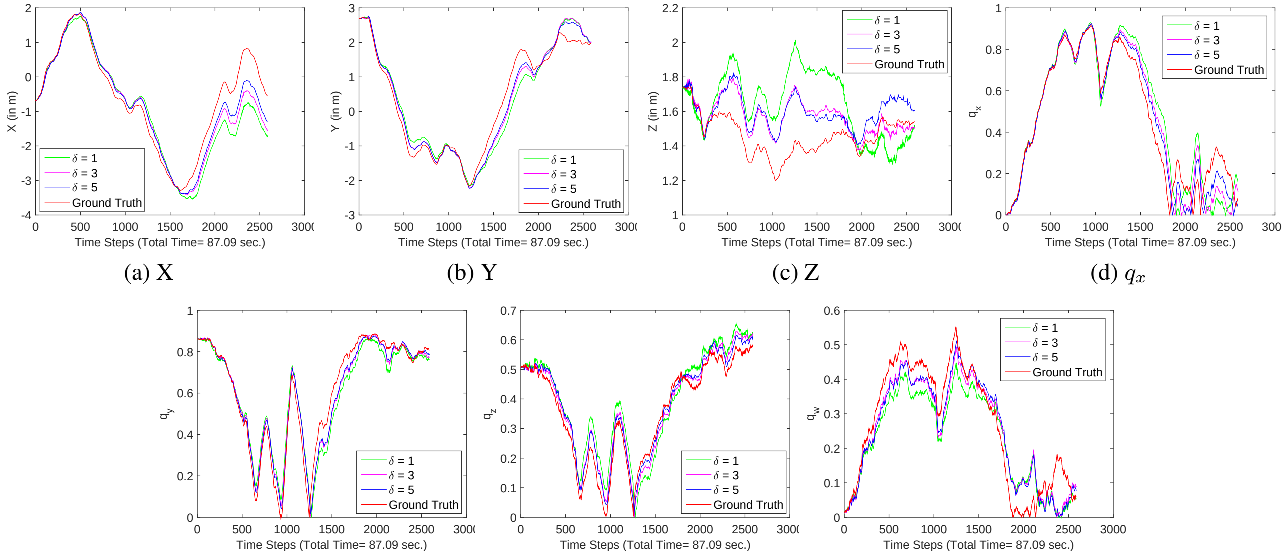
[4] Z. Zhang. Iterative point matching for registration of free-form curves. 1992.

[4] Please note that the experiments are done using euler angles, but as the ground truth is in terms of quaternions, conversion from euler to quaternions have been done for comparison



(a) (b)

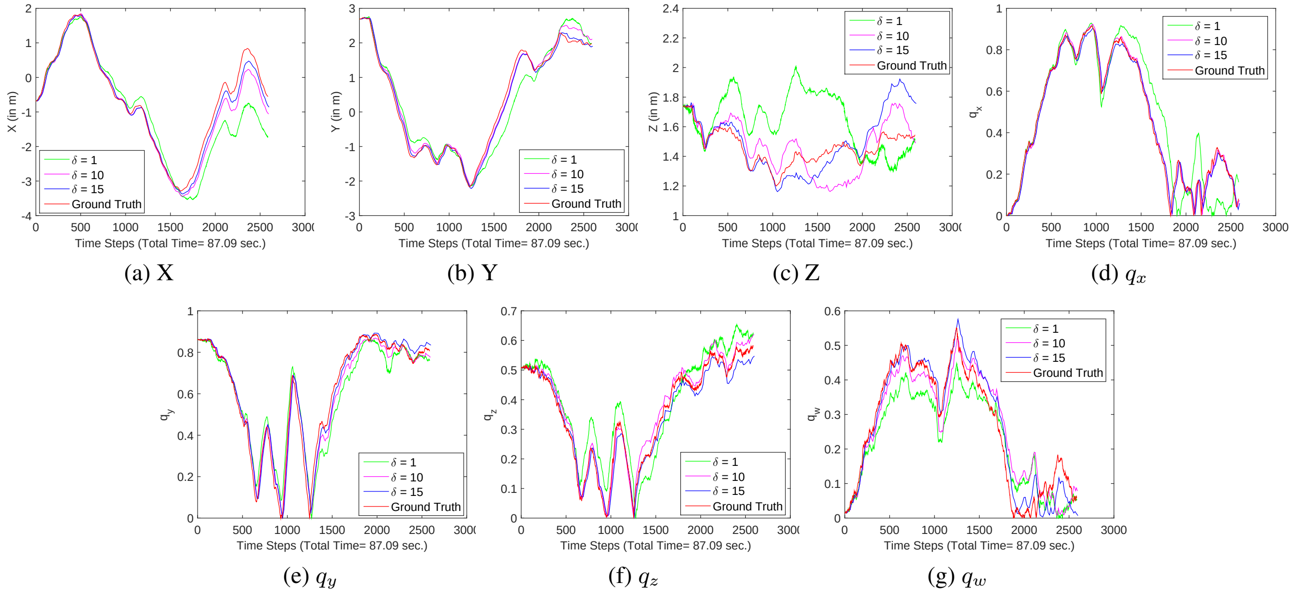
Figure 2: (a) presents the sensitivity of the RANSAC based method on the interval (δ) between the pair of frames used to estimate the transformation matrix. E.g. for $\delta = 3$, frames $1-4, 4-7, \ldots$ are used. (b) presents the comparison of the RANSAC based method (with $\delta = 15$) with ICP (with $\delta = 5$)



(a) X (b) Y (c) Z (d) q_x

(e) q_y (f) q_z (g) q_w

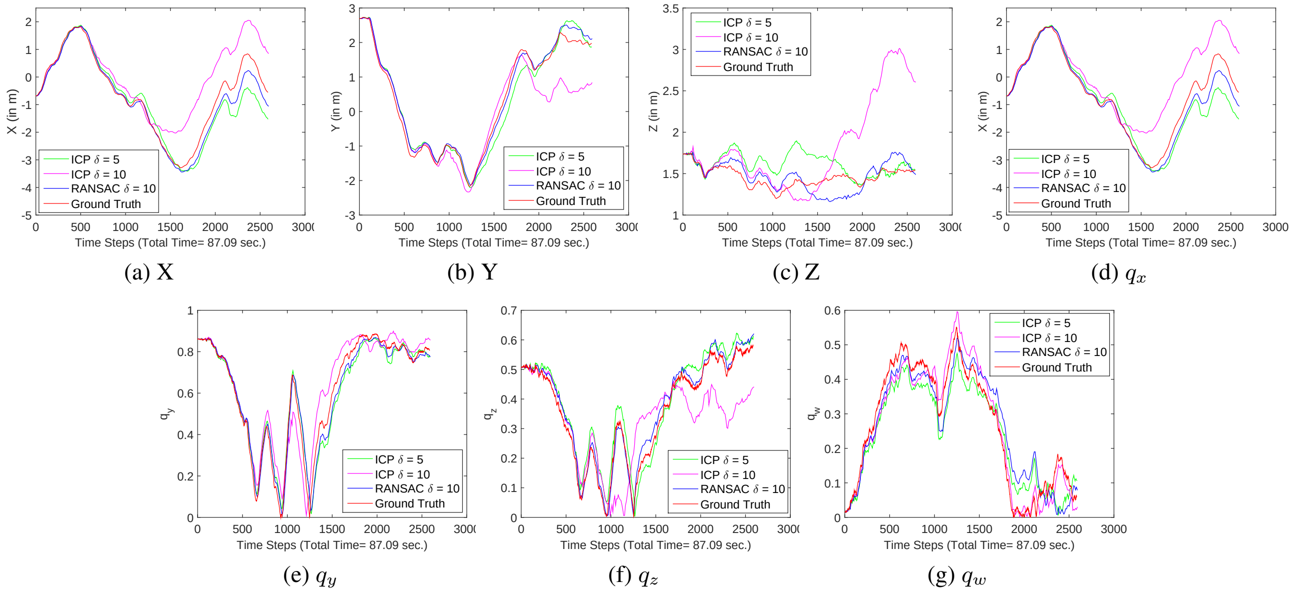
Figure 3: This figure presents the dependence on the interval ($\delta = 1, 3, 5$) between the pair of frames used to estimate the camera pose. E.g. for $\delta = 3$, frames $1-4, 4-7, \ldots$ are used.



(a) X (b) Y (c) Z (d) q_x

(e) q_y (f) q_z (g) q_w

Figure 4: This figure presents the dependence on the interval ($\delta = 1, 10, 15$) between the pair of frames used to estimate the camera pose. E.g. for $\delta = 3$, frames $1-4, 4-7, \dots$ are used.



(a) X (b) Y (c) Z (d) q_x

(e) q_y (f) q_z (g) q_w

Figure 5: This figure presents the comparison of ICP with the RANSAC based method presented in this report. Results include two executions of ICP with $\delta = 5, 10$ and RANSAC with $\delta = 10$ for comparison. E.g. for $\delta = 5$, frames $1-6, 6-11, \dots$ are used to estimate the camera pose.