Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Robotic Systems Lab
Prof. Dr. Marco Hutter

Semester Thesis

Handheld Augmented Reality for Robotic Excavators

Autumn Term 2023

Supervised by:
Dominic Jud
Ryan Luke Johns

Author:
Lasse Fierz

Declaration of Originality

I hereby declare that the written work I have submitted entitled

Robot Localization using Visual Features on LiDAR Data

is original work which I alone have authored and which is written in my own words.[1]

Author(s)

Lasse	Fierz

Student supervisor(s)

Julian	Nubert
Shehryar	Khattak

Committee members(s)

Edo	Jelavic
Alexander	Reske

Supervising lecturer

Marco	Hutter

With the signature I declare that I have been informed regarding normal academic citation rules and that I have read and understood the information on 'Citation etiquette' (`https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetiquette.pdf`). The citation conventions usual to the discipline in question here have been respected.

The above written work may be tested electronically for plagiarism.

Wädenswil 26.07.21

Place and date

Signature

[1] Co-authored work: The signatures of all authors are required. Each signature attests to the originality of the entire piece of written work in its final form.

Intellectual Property Agreement

The student acted under the supervision of Prof. Hutter and contributed to research of his group. Research results of students outside the scope of an employment contract with ETH Zurich belong to the students themselves. The results of the student within the present thesis shall be exploited by ETH Zurich, possibly together with results of other contributors in the same field. To facilitate and to enable a common exploitation of all combined research results, the student hereby assigns his rights to the research results to ETH Zurich. In exchange, the student shall be treated like an employee of ETH Zurich with respect to any income generated due to the research results.
This agreement regulates the rights to the created research results.

1. Intellectual Property Rights

1. The student assigns his/her rights to the research results, including inventions and works protected by copyright, but not including his moral rights ("Urheberpersönlichkeitsrechte"), to ETH Zurich. Herewith, he cedes, in particular, all rights for commercial exploitations of research results to ETH Zurich. He is doing this voluntarily and with full awareness, in order to facilitate the commercial exploitation of the created Research Results. The student's moral rights ("Urheberpersönlichkeitsrechte") shall not be affected by this assignment.
2. In exchange, the student will be compensated by ETH Zurich in the case of income through the commercial exploitation of research results. Compensation will be made as if the student was an employee of ETH Zurich and according to the guidelines "Richtlinien für die wirtschaftliche Verwertung von Forschungsergebnissen der ETH Zürich".
3. The student agrees to keep all research results confidential. This obligation to confidentiality shall persist until he or she is informed by ETH Zurich that the intellectual property rights to the research results have been protected through patent applications or other adequate measures or that no protection is sought, but not longer than 12 months after the collaborator has signed this agreement.
4. If a patent application is filed for an invention based on the research results, the student will duly provide all necessary signatures. He/she also agrees to be available whenever his aid is necessary in the course of the patent application process, e.g. to respond to questions of patent examiners or the like.

2. Settlement of Disagreements

Should disagreements arise out between the parties, the parties will make an effort to settle them between them in good faith. In case of failure of these agreements, Swiss Law shall be applied and the Courts of Zurich shall have exclusive jurisdiction.

Wädenswil 26.07.21
Place and date

Signature

Contents

Preface

The central aspiration of this work is to embrace the progress of a known technology and to attempt to contribute a small part towards understanding it and using it better in the future. I feel tremendously lucky to be able to work on a topic that seems so meaningful to me in addition to being one of the most interesting I have ever been introduced to.

As interesting and fulfilling this field is, it is also very challenging and I would not have been able to achieve the work that I did without the loving support of my family as well as the guidance of my supervisors! For this I would like to thank you!

Abstract

My endeavour in this thesis was to combine state-of-the-art computer vision methods with the new possibilities that the denser LiDAR sensors provide us with in order to achieve motion estimation. To do so I made use of detected 2D visual features on projected LiDAR data in order to establish point correspondences on subsequent frames. These matches could then be used for the closed form solution to solve the point cloud alignment problem. I also put emphasis on researching different feature extraction and descriptor methods (ORB, BRISK, KLT..) and the comparison of their performance on the projections of the different kinds of complementary data considered.

Chapter 1

Introduction

Motion estimation is a well known problem in the robotics world and has been worked on by many brilliant people over a lot of time. Traditionally cameras were used to estimate the robots position however also LiDAR sensors are an option. Now with the newest generation of LiDAR sensors we have a tool at our hands that provides us with scans of 128 pixels vertically distributed and perceives a wider field of view which enables much more dense projections of the point cloud data.
A challenge when using LiDAR data at real-time has always been the vast amount of information that the processor has to deal with leading to a necessity of downsampling. This problem is even enhanced for these denser scans which would have to be downsampled even more in order to achieve real-time performance and thus lose a lot of valuable data.
An idea to counteract this problem while preserving the advantages of the newer generation LiDAR scans is using 2D methods. It is fair to say that 2D computer vision methods are better researched and are faster than iterative 3D point handling methods. So an approach could be to consider the scanned dense 3D data as 2D data through projections in order to then apply fast and refined 2D methods to establish point correspondences and thus be able to apply the closed form solution to the motion estimation problem.

Chapter 2

Related Work

The work related to this thesis can be divided into two categories:
First there is the traditional 2D computer vision aspect to it for which I can't reference a specific work as the field is just too broad. However some crucial mentions are the different feature methods that I considered in this work (ORB[1], BRISK [2] and KLT[3]) as well as the outlier rejection procedure RANSAC[4].
The second part concerns the 3D side of the paper being state of the art LiDAR usage for motion estimation:
LiDAR as a tool is of course no new idea and many ingenious people have already ventured into this field and refined methods to work with the 3D data that LiDAR provides us with. The traditional procedure for achieving motion estimation using LiDAR is point cloud registration and there have been a lot of papers published about this idea. One that I would like to point out is the paper of Pomerleau et al. [5] which summarizes the ICP algorithm (Iterative Closest Point) as well as certain usage cases.
An alternative procedure to estimate the motion and construct a map of the surroundings at run time is LOAM [6] which is built around the idea of splitting the two algorithms up into the odometry and the mapping part. For the odometry part the detected features are divided into planar patches and sharp edge lines which are then used to establish correspondences and thus achieve motion estimation. The mapping process makes use of the iterative scans as well as the transformations each step in order to build the permanent map in the world frame. This map in turn can be consulted in order to achieve much more accurate motion estimation.

This bachelor thesis however was built on the idea of projecting dense point clouds of newer LiDAR scans onto planes and performing state of the art 2D CV methods on the projections as opposed to applying computationally expensive alignment methods. I thus used the best of both worlds by achieving run time performance without neglecting a significant amount of information through downsampling the point clouds.

[1] Oriented FAST and Rotated BRIEF [1]
[2] BRISK: Binary Robust Invariant Scalable Keypoints [2]
[3] Lucas Kanade Tracking [3]
[4] RANdom SAmple Consensus[4]
[5] A Review of Point Cloud Registration Algorithms for Mobile Robotics [5]
[6] LiDAR Odometry and Mapping [6]

With great probability the first paper published about this new way of working with LiDAR data is the paper of Shan et al. [7] In their work they used this approach in order to extract ORB features each scan and to build up a BoW database which they queried in order to find matches with later extracted features to determine loop closures. In comparison to their work I used this underlying method in order to achieve motion estimation considering just two subsequent scans as well as different descriptors and complementary types of data.

[7] Robust place recognition using an imaging lidar [7]

Chapter 3

Method

My approach consists of the following stages:

- LiDAR Dataset
  Throughout this project I worked with an Ouster OS1-128 scan courtesy of the paper of Shan et al. [1] More specifically it is a handheld outside dataset characterized by urban structures like houses, vegetation and cars. As the sensor is carried throughout the whole scan duration it undergoes significant altitude change.
- Image Projection
  Different types of complementary point data from the input scan were used to perform projections onto an image plane.
- Feature Extraction
  The projections were treated like ordinary images on which features were extracted.
- Feature Matching
  The extracted features went on to be matched on subsequent frames for point correspondences.
- Outlier Rejection
  Bad matches were neglected to improve the performance.
- Motion Estimation
  This was the final pursuit of the pipeline. Through the correspondences acquired through 2D CV techniques a closed form solution could be applied for the alignment of two subsequent point clouds in order to estimate the motion in between.
- Comparison of Feature Methods and Complementary Data
  Throughout the whole project I considered different feature methods to work with as well as the different complementary data types on which to apply the designed pipeline. The comparison of the possible combinations succeeded along the indiviudal steps of the implementation.

[1] [7]

Chapter 4

Results

4.1 Ground Truth

As there wasn't a GT for my dataset I considered the very accurate and map based Lio-Sam[1] estimation as the GT for this work.
I also considered LOAM[2] for another comparison and very importantly as a third and most meaningful alternative I chose the LOAM method without the map procedure as well. This is because my approach is scan based and does not build a map which of course enhances accuracy significantly. So comparing to a scan based method is the only fair comparison.

4.2 Stepwise Results

4.2.1 Comparison of Separated Components

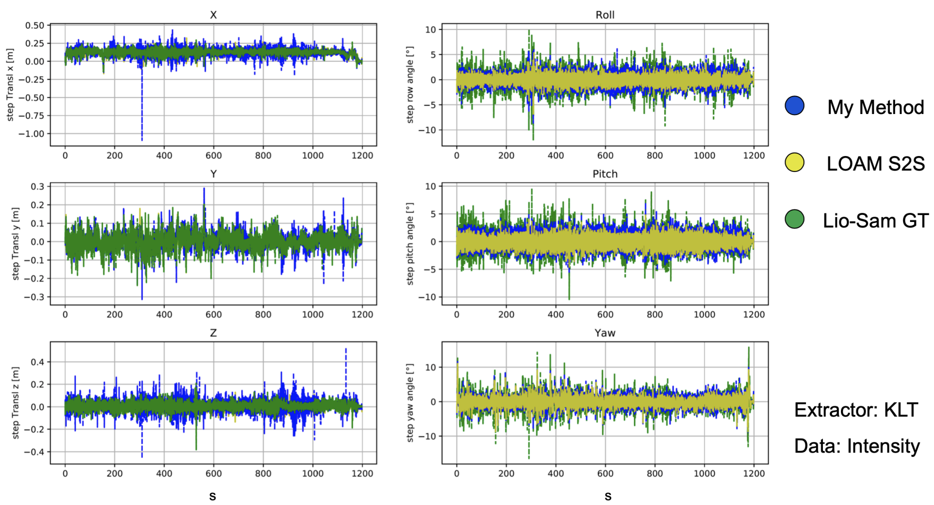

Figure 4.1: Step comparison

In fig. 4.1 we can see the step comparison of the three directions as well as the rotation angles.

[1]LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping [8]
[2]LiDAR Odometry And Mapping [6]

We can also consider the errors at each iteration depicted in the following fig. 4.2

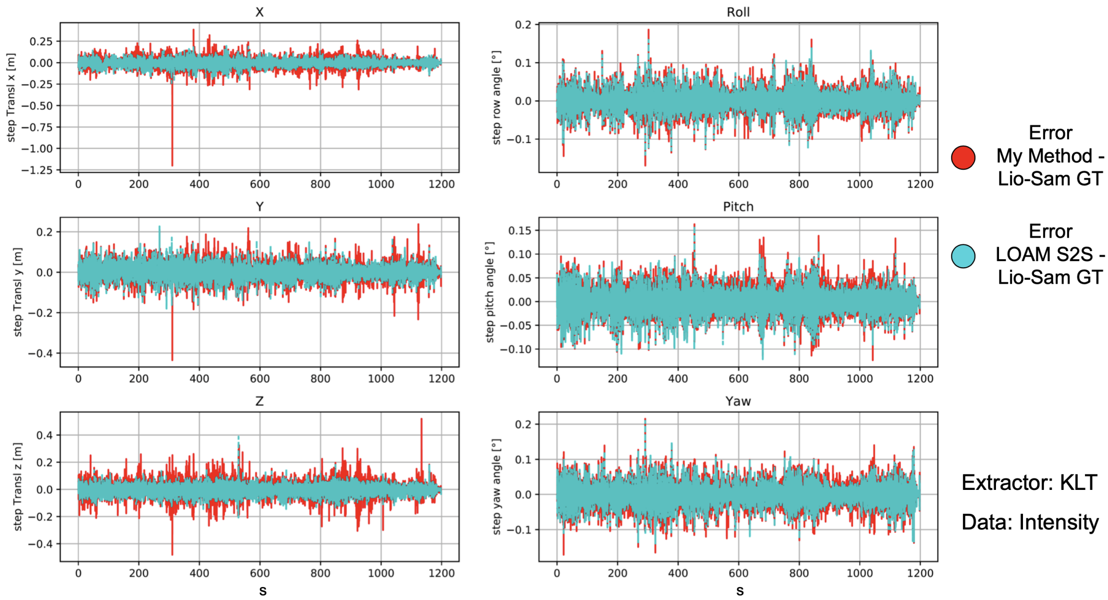

Figure 4.2: Step Error comparison

As we can see in the figures fig. 4.1 and fig. 4.2 there are outliers in the method. However both the step error of this papers method as well as the iterative error of the scan based LOAM method yield values in the same range.

4.2.2 Average Step Performance Comparison

Mean iterative error and standard deviation thereof regarding GT over the whole dataset:

Errors	**My Method**		**LOAM S2S**	
	Mean	STD	Mean	STD
X[m]	-0.004	0.056	-0.002	0.035
Y[m]	0	0.040	0.005	0.028
Z[m]	-0.001	0.058	-0.004	0.032
Roll[°]	0	0.024	0	0.021
Pitch[°]	0	0.024	-0.001	0.020
Yaw[°]	0	0.024	0	0.023

Table 4.1: Average Step Error Comparison

The data displayed in table 4.1 was achieved using the KLT method on intensity data. Green values are the best achieved mean while light

blue indicates the best standard deviation.
When evaluating the numbers we see for my method smaller mean errors on average while LOAM scan 2 scan has smaller deviations from the slightly larger mean error.

4.3 Global Results

4.3.1 Comparison of Separated Components

Comparing the individual pose components:

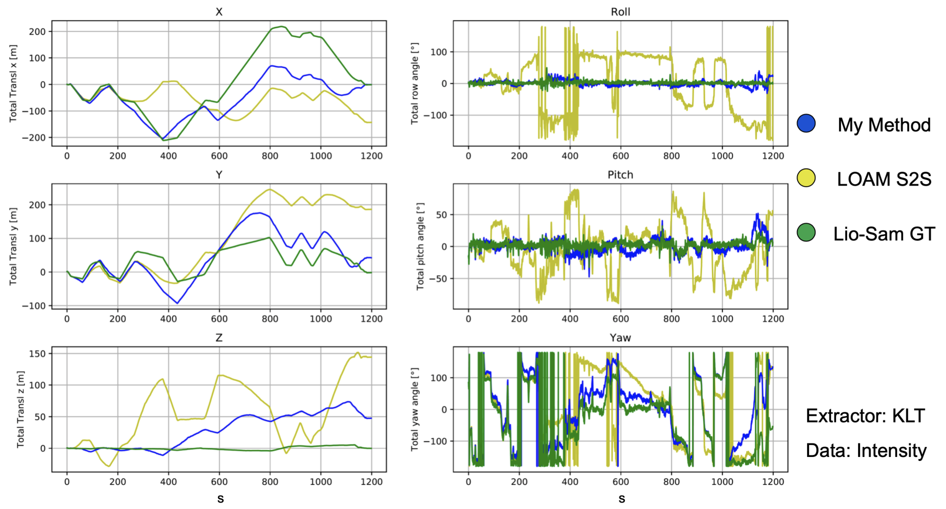

Figure 4.3: Pose comparison

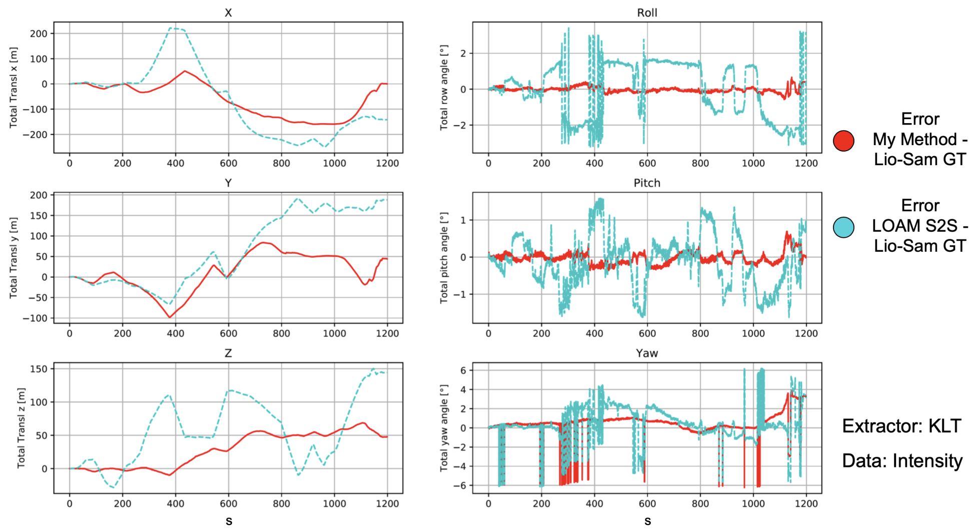

Figure 4.4: Pose error comparison

Here in fig. 4.3 and fig. 4.4 we can see substantially better global performance of this works method especially regarding the angular development.

4.3.2 Comparison of Global Error

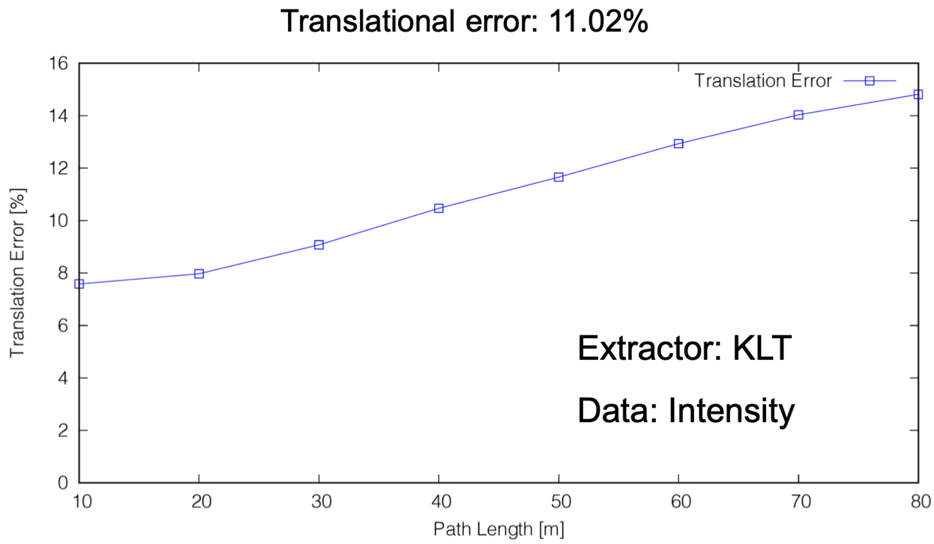

Figure 4.5: My methods translational drift

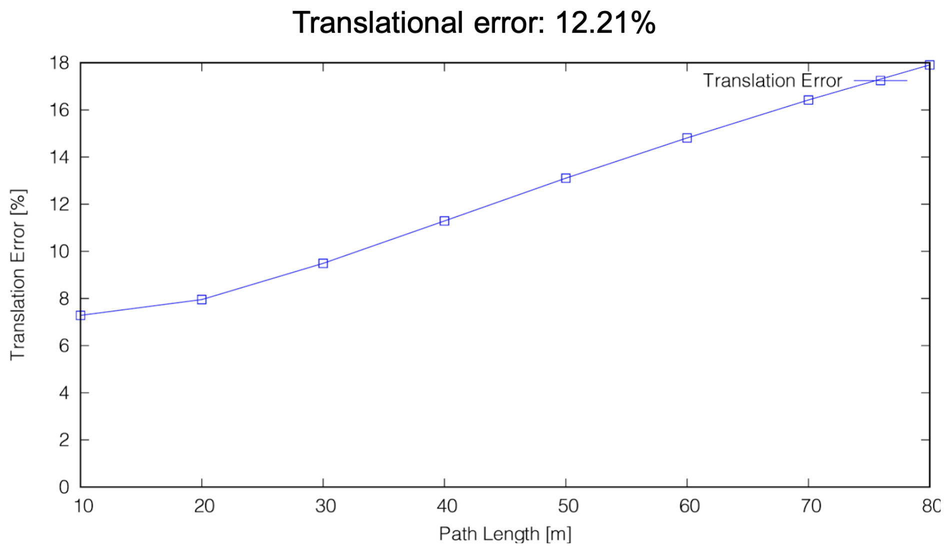

Figure 4.6: LOAM translational drift

To account for the drift of the motion estimation a comparison was done after predetermined distance intervals. Here in fig. 4.5 to fig. 4.6 are the values for the intervals from 10 to 80 meters. It has to be noted that 80 meters is quite a distance and the results from that margin onwards might be slightly deceiving compared to more solid smaller intervals.
Averaged over all interval sizes my methods performs a little better as can be seen from the averaged translational error denoted at the top.

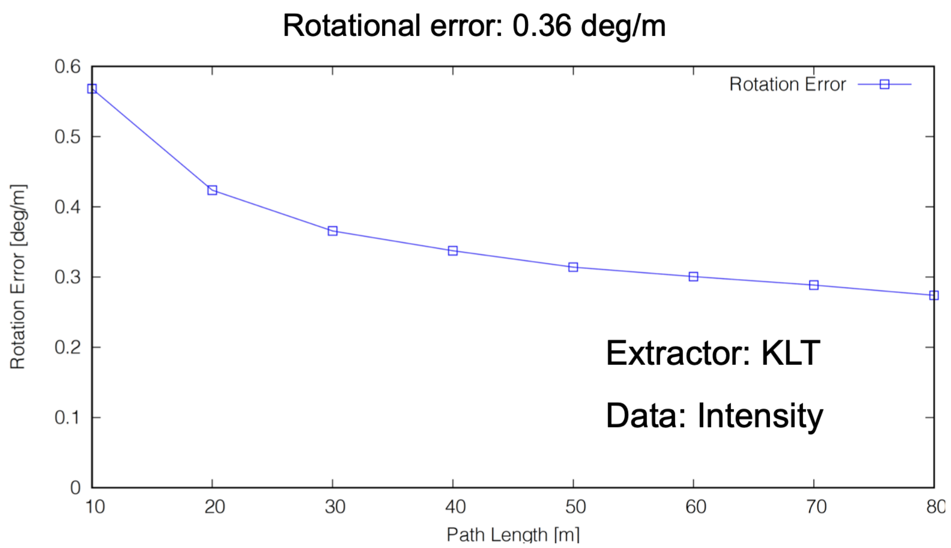

Figure 4.7: My methods angular drift

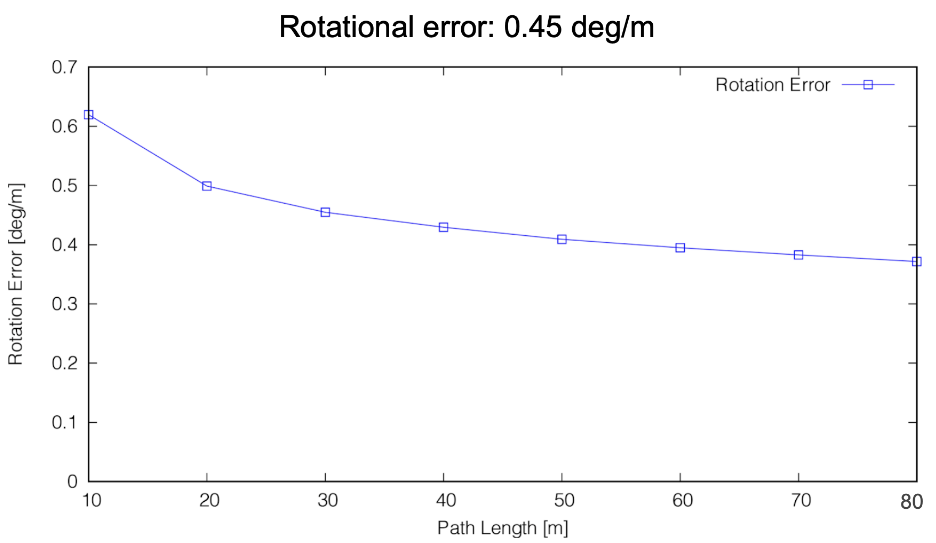

Figure 4.8: loam angular drift

A similar result can be drawn from the angular development. This papers method shows with 0.35 degrees/meter a better performance than the LOAM scan to scan method.

4.3.3 Comparing Estimated Trajectories

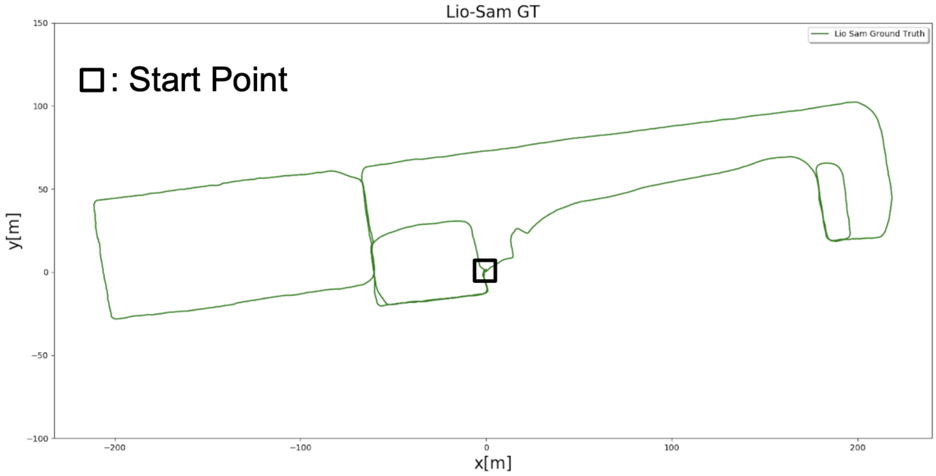

Figure 4.9: GT trajectory

Indicated on the plot above is the datasets trajectory estimated by Lio-Sam. This will again be considered the ground truth.

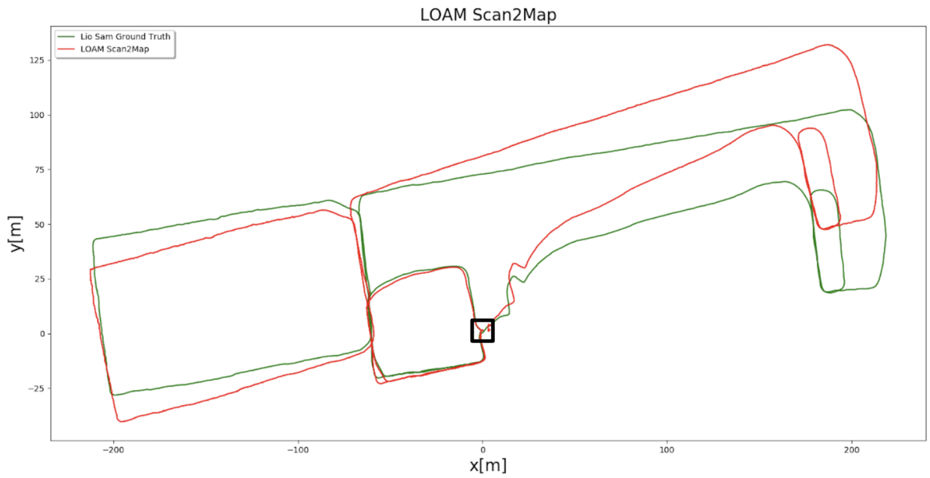

Figure 4.10: LOAM scan to map trajectory

As aforementioned the LOAM method performs very well as it is also map based and thus really accurate.

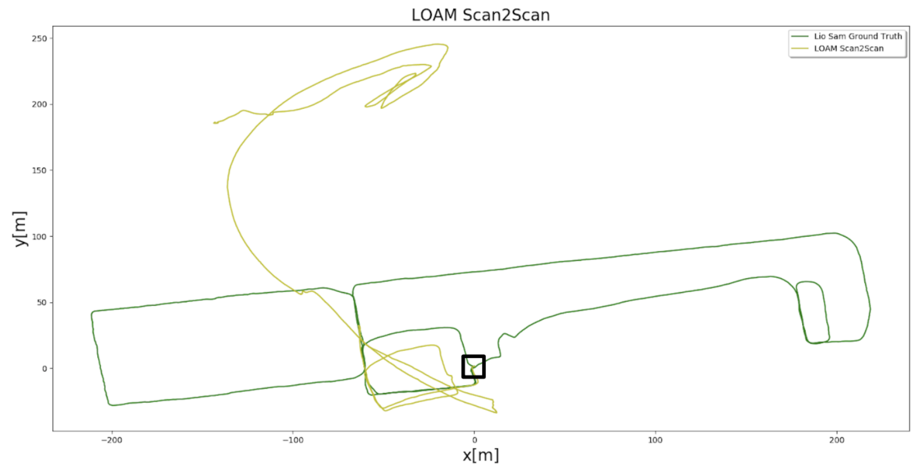

Figure 4.11: LOAM scan to scan trajectory

Considering the solely scan based LOAM estimation it looks quite poor.

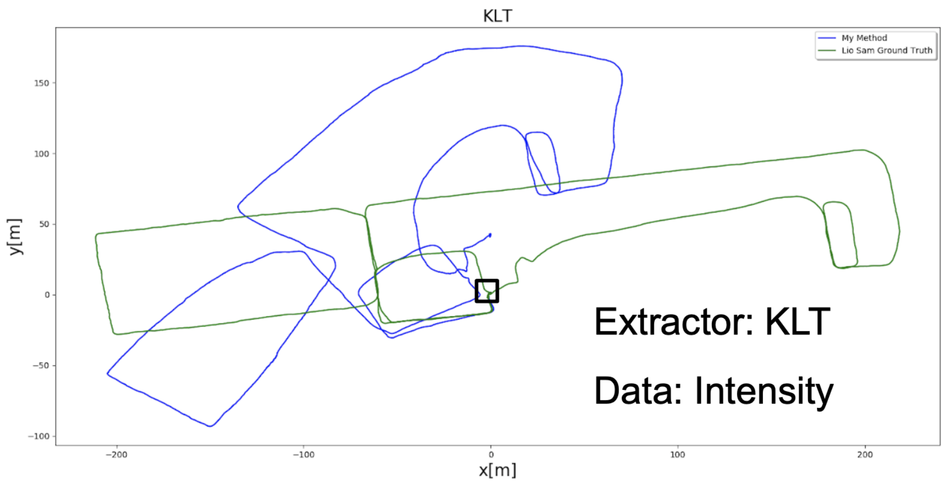

Figure 4.12: This works methods trajectory

When looking at method from this paper it does accumulate quite some drift but performs considerably better than the scan based LOAM implementation. For additional performance results of this method consult section 4.6

4.4 Final Comparison of Feature Methods

In this section I will perform a comparison of the feature methods to the point of a possible verdict. For in depth data about each respective method consult appendix A

4.4.1 Local Step Comparison – feature methods

	ORB		**BRISK**		**KLT**	
Errors	Mean	STD	Mean	STD	Mean	STD
X[m]	-0.004	0.056	0.004	0.219	-0.003	0.048
Y[m]	0	0.040	0	0.117	0.001	0.035
Z[m]	-0.001	0.058	-0.001	0.097	0	0.046
Roll[°]	0	0.024	0.001	0.049	0	0.025
Pitch[°]	0	0.024	0	0.028	0	0.024
Yaw[°]	0	0.024	0	0.048	0	0.029

Table 4.2: Average Step Error Comparison for Feature Methods

In table 4.2 we can see the iterative performance of the feature methods.
ORB and KLT seem to perform similarly while BRISK falls off a little regarding the standard deviation.

4.4.2 Trajectory Comparison – feature methods

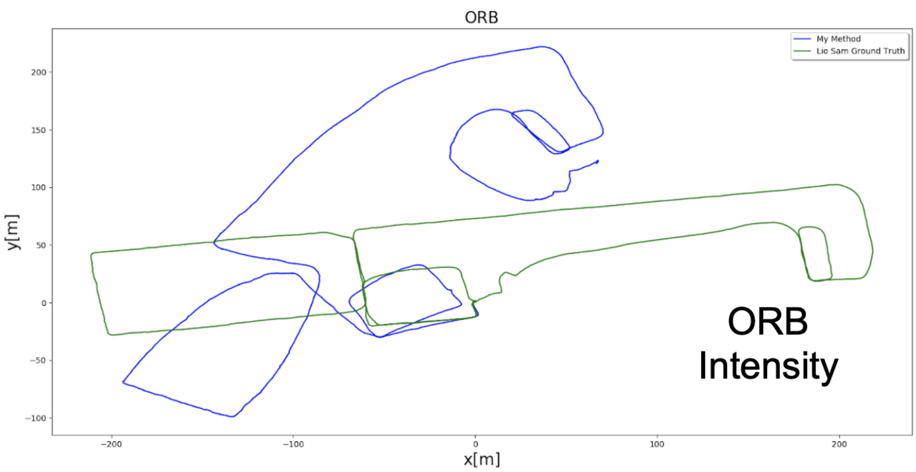

Figure 4.13: ORB trajectory

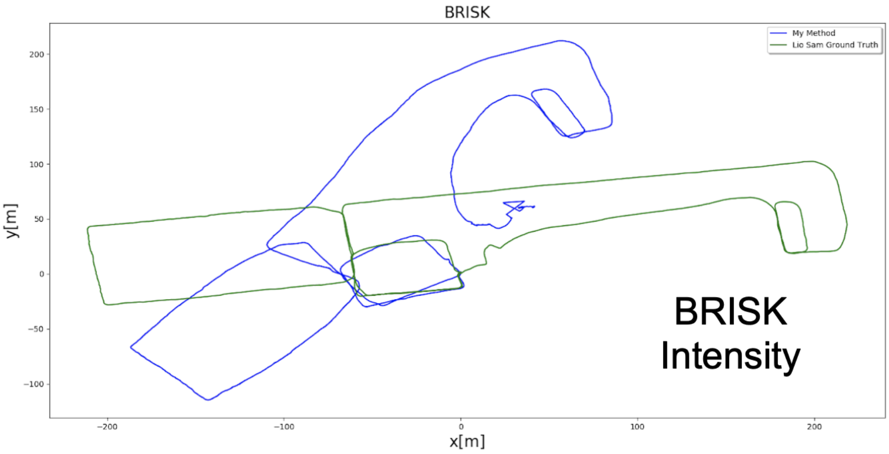

Figure 4.14: BRISK trajectory

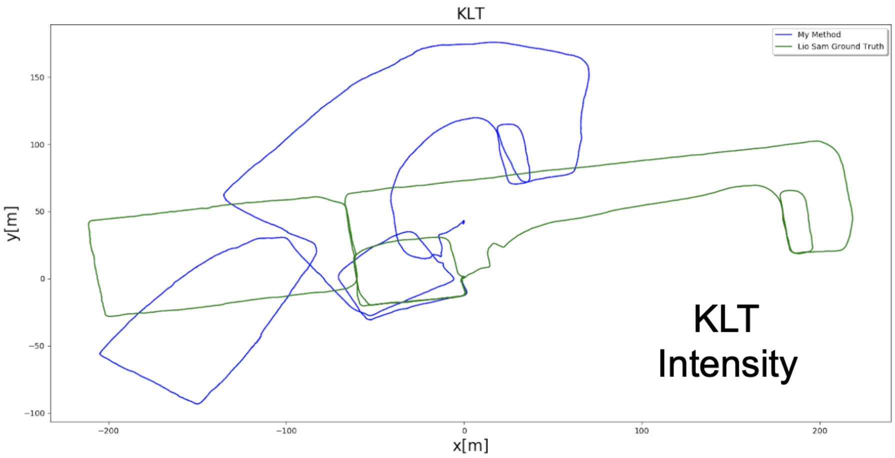

Figure 4.15: KLT trajectory

4.4.3 Drift Comparison – feature methods

I also compared the more promising methods ORB and KLT regarding the accumulated drift:

Translational Drift - feature methods

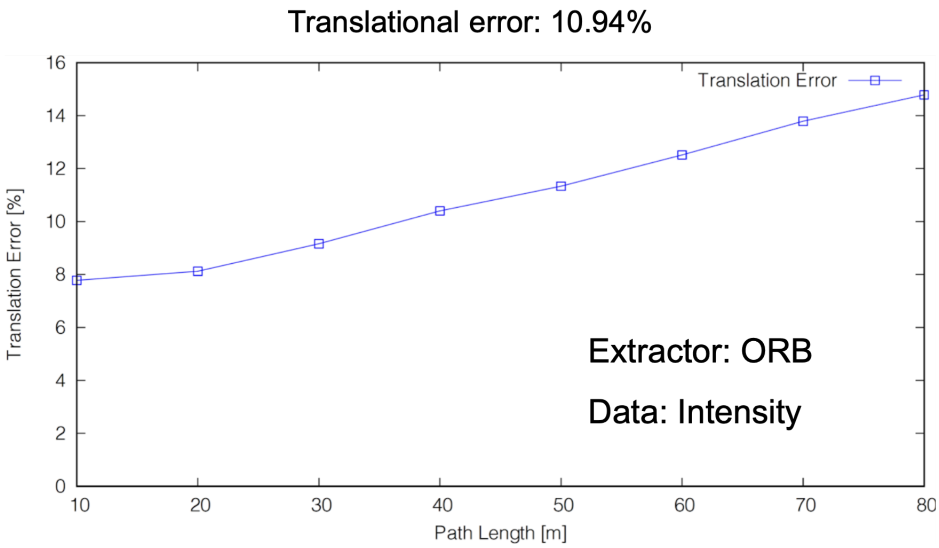

Figure 4.16: ORB drift translation

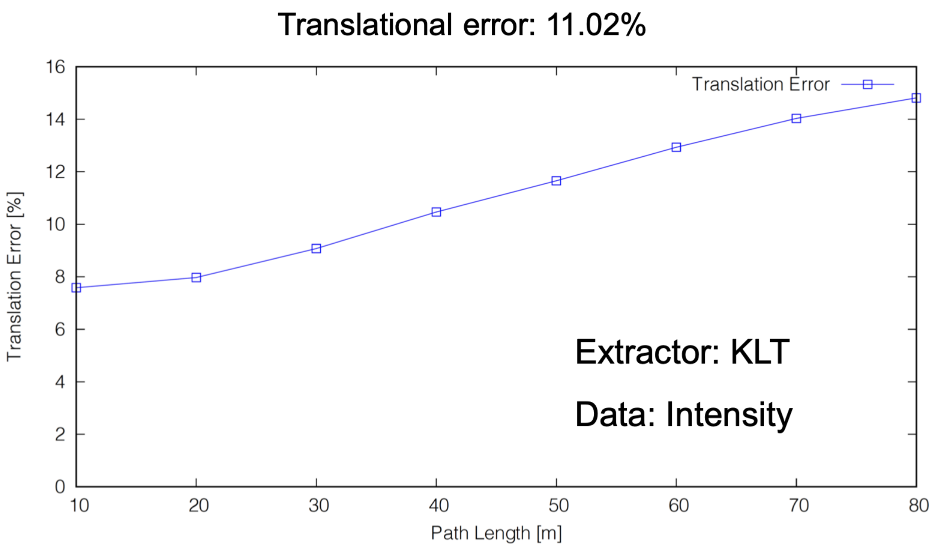

Figure 4.17: KLT translational drift

Angular Drift - feature methods

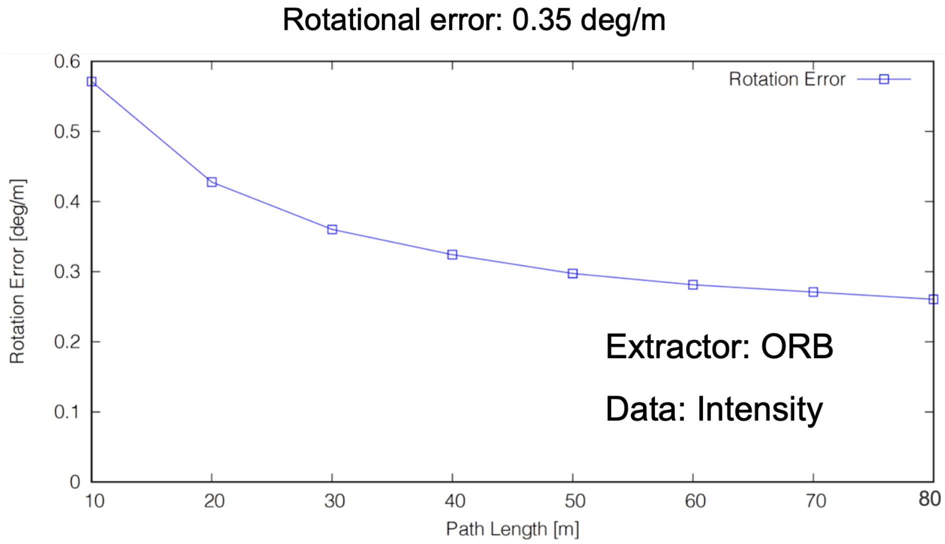

Figure 4.18: ORB angular drift

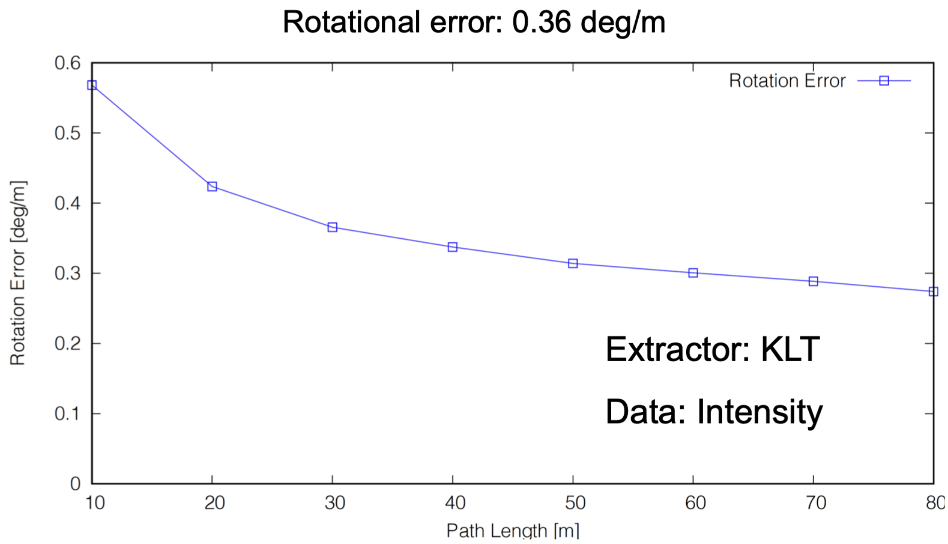

Figure 4.19: KLT angular drift

As we can see on the drift plots fig. 4.16 to fig. 4.19 ORB performs slightly better than KLT for the considered dataset.

4.4.4 Verdict Feature Methods

Starting of with the step comparison ORB and KLT indicate a slightly better performance than BRISK.
With the additional consideration of BRISKs comparatively poor TP rate and higher computational cost I would chose ORB or KLT over BRISK for the endeavor pursued in this work.
Then for the comparison of ORB and KLT we can see that ORB performs a little bit better considering the global drift error. However we have to consider the fundamental difference in their procedure. While ORB is dependent on the extraction of points at each iteration KLT can make use of previously detected points. This can be an advantage especially in feature-scarce environments. This is shown well in the additinal performance results section section 4.6. So for a final verdict in this comparison I would use ORB on feature rich environments while KLT is more consistent in repetitive and feature scarce surroundings.

4.5 Final Comparison of Complementary Data

4.5.1 Local Step Comparison – projection types

Errors	**Intensity**		**Ambient**		**Range**	
	Mean	STD	Mean	STD	Mean	STD
X[m]	-0.004	0.056	-0.007	0.060	-0.002	0.102
Y[m]	0	0.040	-0.001	0.042	-0.008	0.170
Z[m]	-0.001	0.058	0	0.060	0	0.098
Roll[°]	0	0.024	0	0.024	0.001	0.072
Pitch[°]	0	0.024	0	0.024	0	0.039
Yaw[°]	0	0.029	0	0.029	0	0.040

Table 4.3: Average Step Error Comparison for Complementary Data

In table 4.3 intensity performs best directly followed best by the ambient data. Range falls off considering the standard deviation.

4.5.2 Trajectory Comparison – projection types

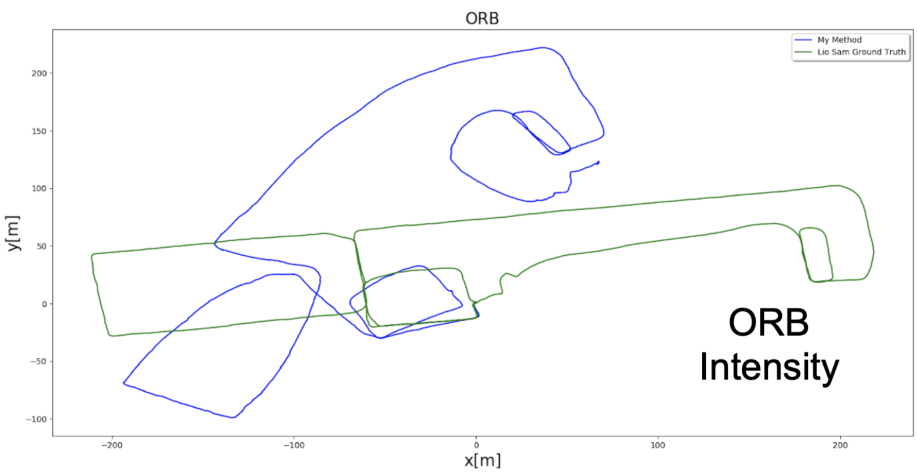

Figure 4.20: ORB trajectory

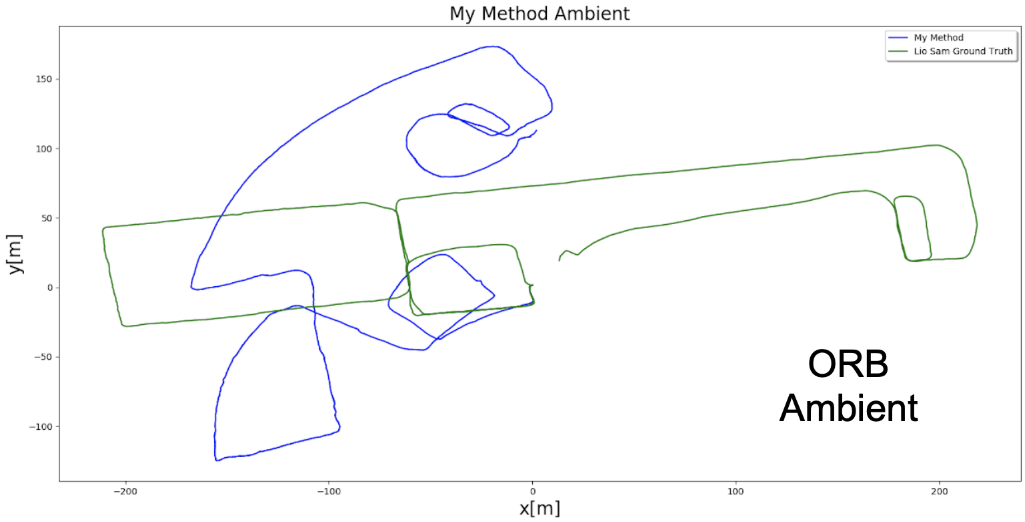

Figure 4.21: Ambient trajectory

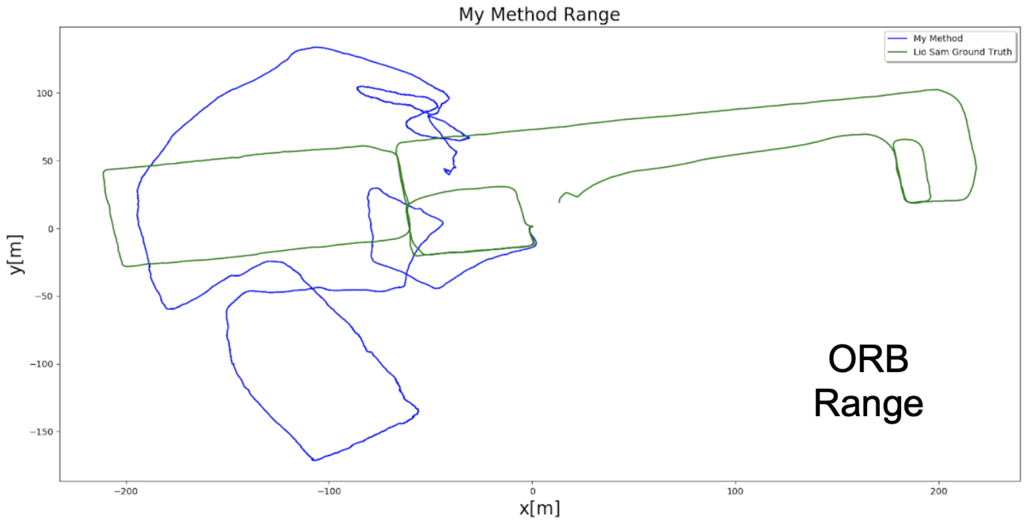

Figure 4.22: Range trajectory

4.5.3 Drift Comparison – projection types

Analogously the accumulated drift comparison for the two more promising data types intensity and ambient:

Translational Drift - projection types

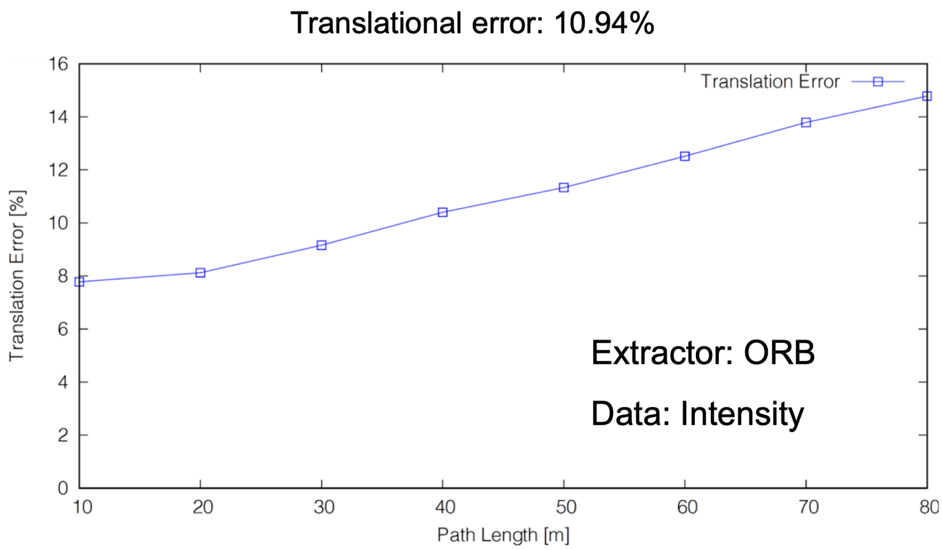

Figure 4.23: Intensity drift translation

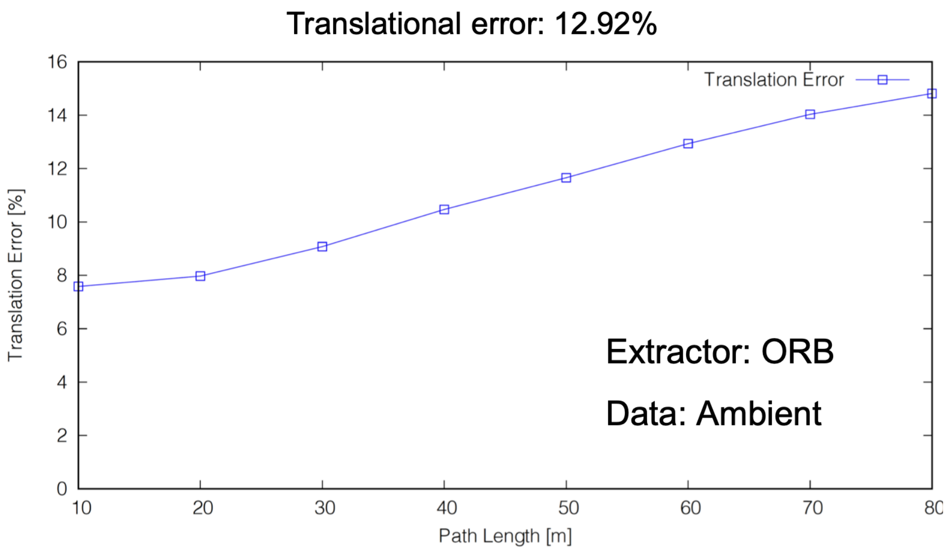

Figure 4.24: Ambient drift translation

Angular Drift - projection types

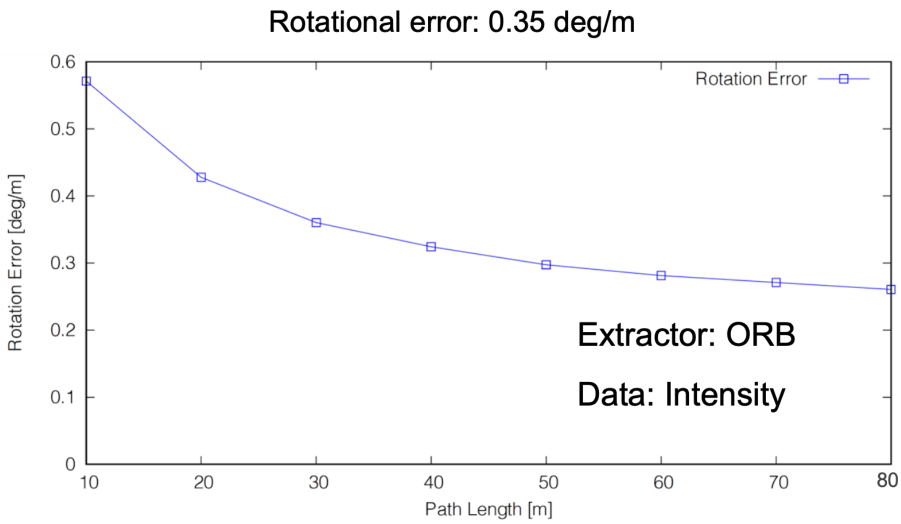

Figure 4.25: Intensity angular drift

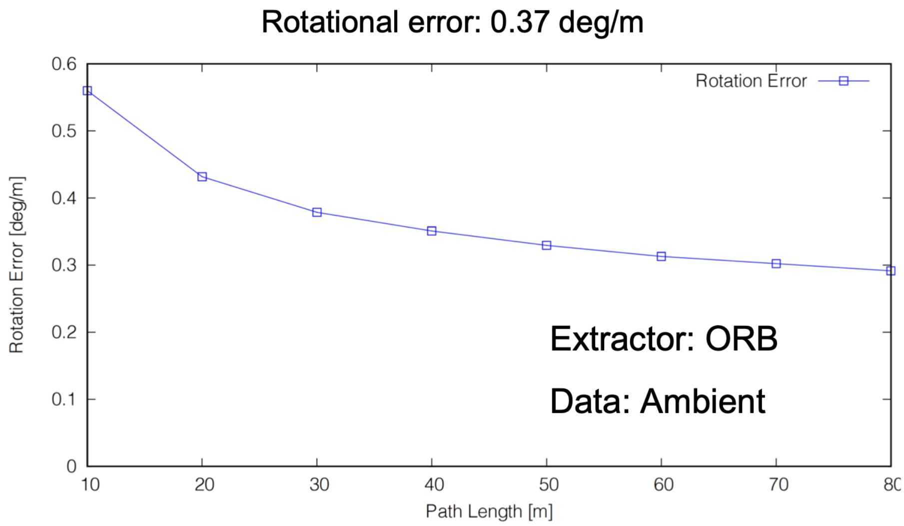

Figure 4.26: Ambient angular drift

On the drift plots fig. 4.23 to fig. 4.26 we see the intensity data performing better than ambient.

4.5.4 Verdict Complementary Data

For this comparison the story is very similar for each stage in the progression. Intensity and ambient data perform equally well as the data source with a small advantage of intensity data. Range however falls off heavily. This can be seen throughout the whole pipeline. Fewer matches were built, a smaller TP rate was detected, bigger iterative as well as global errors could be found and a worse trajectory estimation resulted.

The small gap between intensity and ambient data becomes noticeable when considering the laser intensities light independent nature. Therefore my choice for the optimal data source is intensity while keeping the ambient data in mind. Of course a combination of the data sources (and feature methods) considered would be a better solution still but this work revolved exclusively around individual performance comparison.

4.6 Additional Performance Results

4.6.1 Map Construction using found Transformation

With the pose estimation and the iterative scans we can build a map as depicted in fig. 4.27 and fig. 4.28.

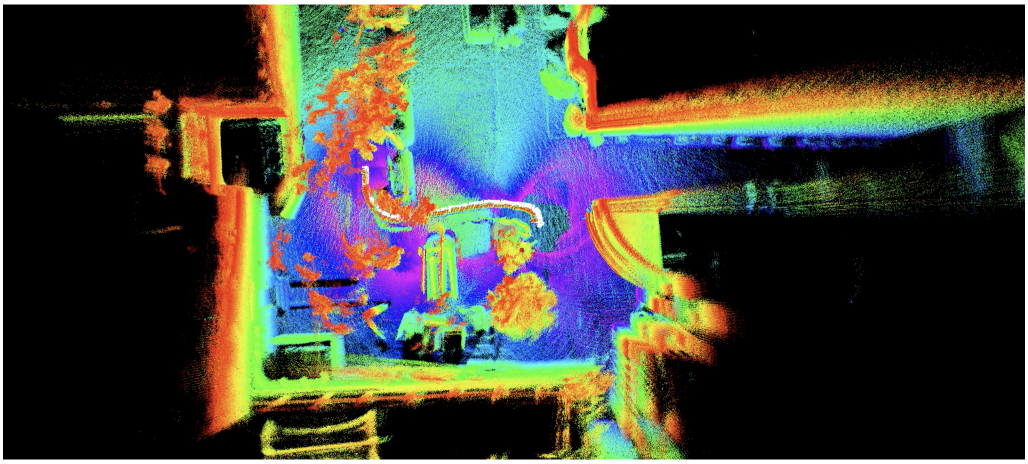

Figure 4.27: Local mapping performance

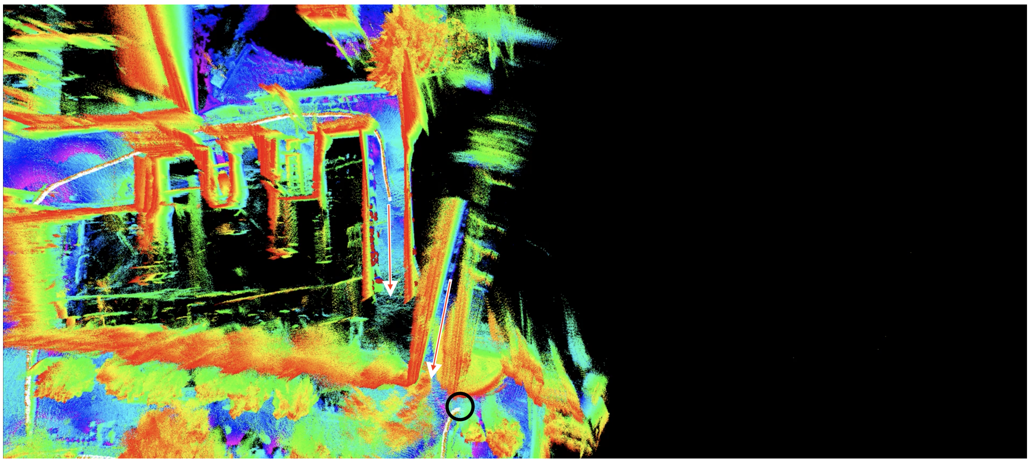

Figure 4.28: Mapping drift after walking one block

As we can see in fig. 4.27 locally the estimation is really accurate leading to a detailed map of the surroundings.
After having gone around the building block in the dataset however the mapping process shows accumulated drift. (fig. 4.28)
For a perfect transformation the red-white arrows should be aligned and should lead back to the start position indicated with the black circle.

4.6.2 Alternative Dataset 1: Indoor

For a first alternative to the outside dataset I considered an indoor scan from the same paper[7] as the handheld dataset. In this dataset three laps are walked through an office space with small corridors. Before the third lap however the LiDAR sensor is turned upside down.

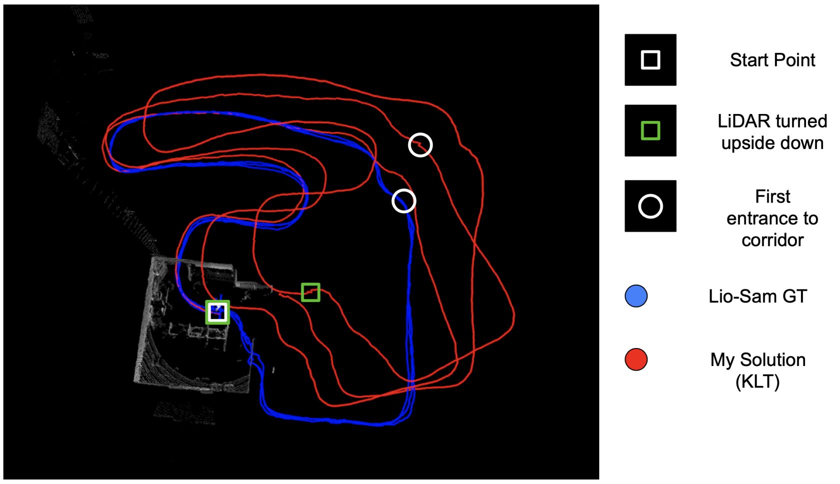

Figure 4.29: Indoor Dataset using KLT

As we can see there is certain drift accumulating but only slowly. Locally the performance seems quite robust. Also interestingly there is no observable decrease in performance after the second loop. (Sensor turning point)
This result was achieved using the KLT method. So far ORB and KLT had performed similarly with ORB having slightly smaller errors. Here for this dataset however we have to work with very little features and repetitive surroundings in the corridors. The impact of this can be seen in fig. 4.30 when using ORB:

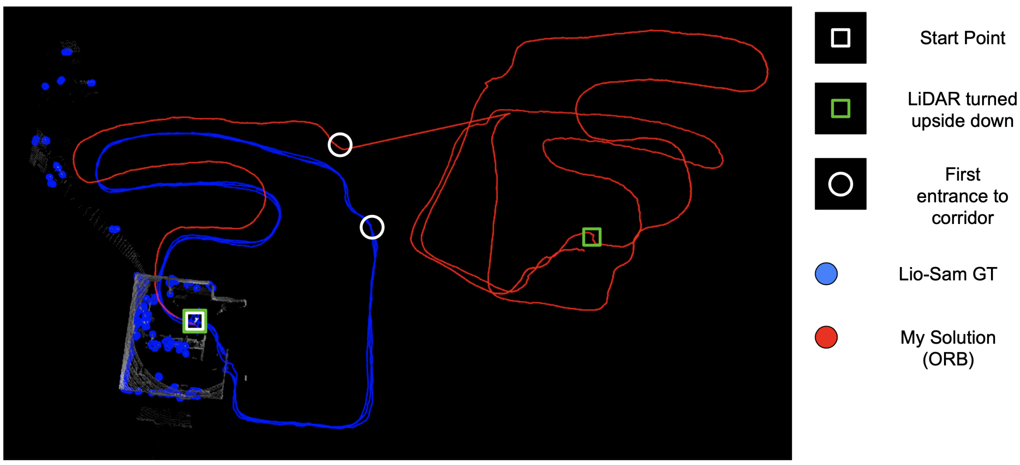

Figure 4.30: Indoor Dataset using ORB

As we can see ORB performs significantly worse than KLT starting at the entrance point of the corridor (indicated with the white circle) this is of course due to the lack of detectable features while KLT can make use of previously detected key points.

4.6.3 Alternative Dataset 2: Excavator

As a second alternative dataset I considered the excavator dataset from the rsl lab.

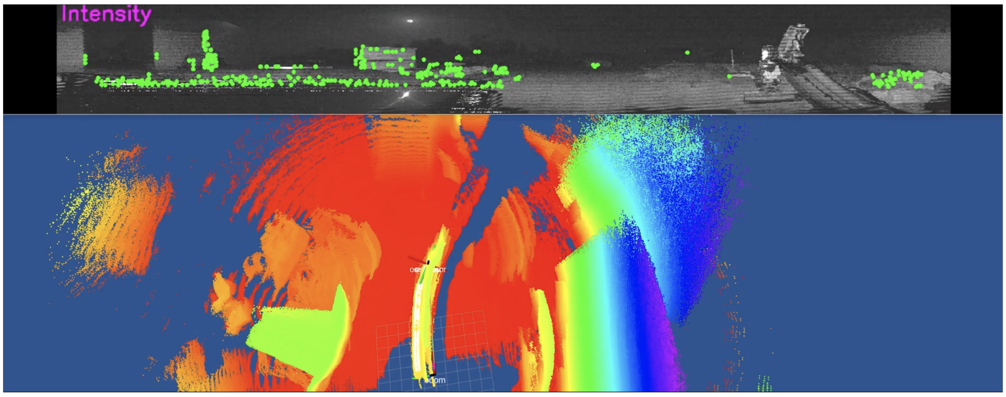

Figure 4.31: Map construction on excavator dataset

As the name lets assume this dataset is from the point of view of an excavator on a very vast landscape with few features that are located far away. In the lower half of this fig. 4.31 we see the map building process analogously to fig. 4.27. Here we see poor performance due to the lack of features and the big distances to them.

Chapter 5

Conclusion

Before concluding this work I would like to mention ?? where I stated my recommendation of an optimal parameter settings used for this pipeline.

5.1 General Performance

In this work I presented a method to achieve run-time motion estimation using modern LiDARs in combination with state of the art 2D CV methods without the need to downsample the scans and thus losing valuable information.
The method presented achieves substantially better results when compared to scan based LOAM. In general very little local drift can be seen whereas globally drift does accumulate.[1]

5.2 Evaluation of Feature Methods

BRISK does detect more points and thus constructs significantly more matches than the other two methods. This is cause for a higher computational cost. However it loses quite a lot of these matches in the outlier rejection process effectively leading to fever filtered matches than the other two methods had and thus a worse TP rate.
The motion estimation performance shows similar insight as the mean error is in the same range as the other methods but BRISKs error deviates stronger from the mean than KLT or ORB.
Finally for the comparison of ORB and KLT it can be said that these methods perform similarly well. ORB has better numbers on feature rich environments while KLT has the advantage of consistency in feature scarce surroundings. (fig. 4.29)

5.3 Complementary Data Comparison

Intensity proved to be a consistent and good performing data source to consider. Ambient data yields similar performance however as it is dependent on the lighting of the surrounding intensity proves to be the best individual choice.
Range falls off quite heavily regarding all phases. Fewer points are extracted, fewer inliers are considered in these correspondences and a high iterative error can be seen compared to the other two data types.

[1]Can be seen clearly in fig. 4.27

5.4 Main Limitations

As I considered a dataset of urban structures like houses at a close distance there were most often substantially many features close by. When considering different datasets however with scarce amounts of features which are further away the performance suffered. This is shown in fig. 4.31.

5.5 Future Work

The method presented shows promising results for the implemented scan based pipeline. The results could however be improved using a combination of the feature methods as well as a combination of the complementary data types. Furthermore considering multiple subsequent frames instead of just two would of course refine the performance significantly.

Bibliography

[1] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," 2011.

[2] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," 2011.

[3] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," 1981.

[4] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with apphcatlons to image analysis and automated cartography," 1981.

[5] F. Pomerleau, F. Colas, and R. Siegwart, "A Review of Point Cloud Registration Algorithms for Mobile Robotics. Foundations and Trends in Robotics," *hal archives*, pp. 1–104, 2015.

[6] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," 2014.

[7] T. Shan, B. Englot, F. Duarte, C. Ratti, and R. Daniela, "Robust place recognition using an imaging lidar," in *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021.

[8] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and R. Daniela, "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 5135–5142.

Appendix A

Additional Plots

A.1 ORB Comparison Plots

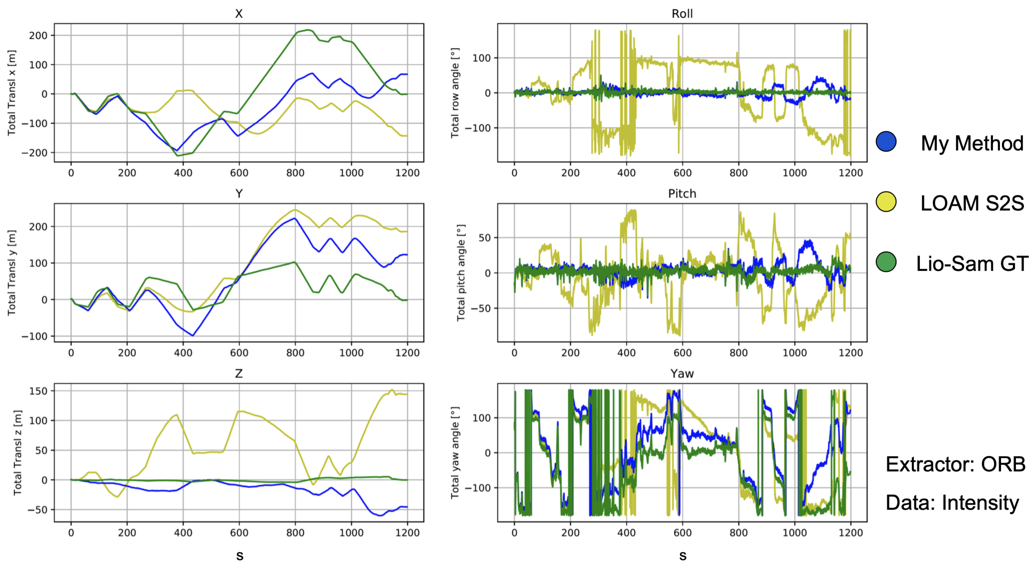

Figure A.1: Pose comparison orb

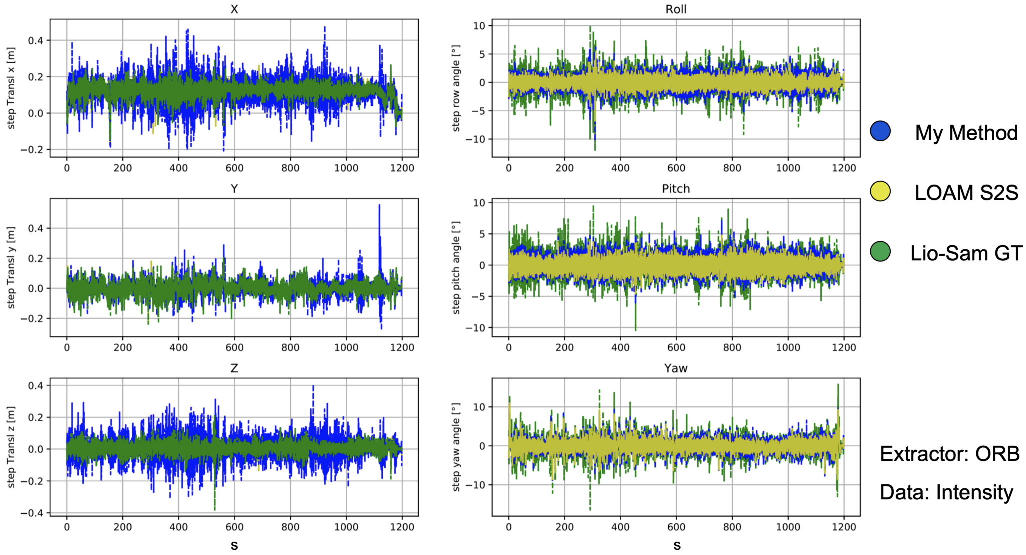

Figure A.2: Step comparison orb

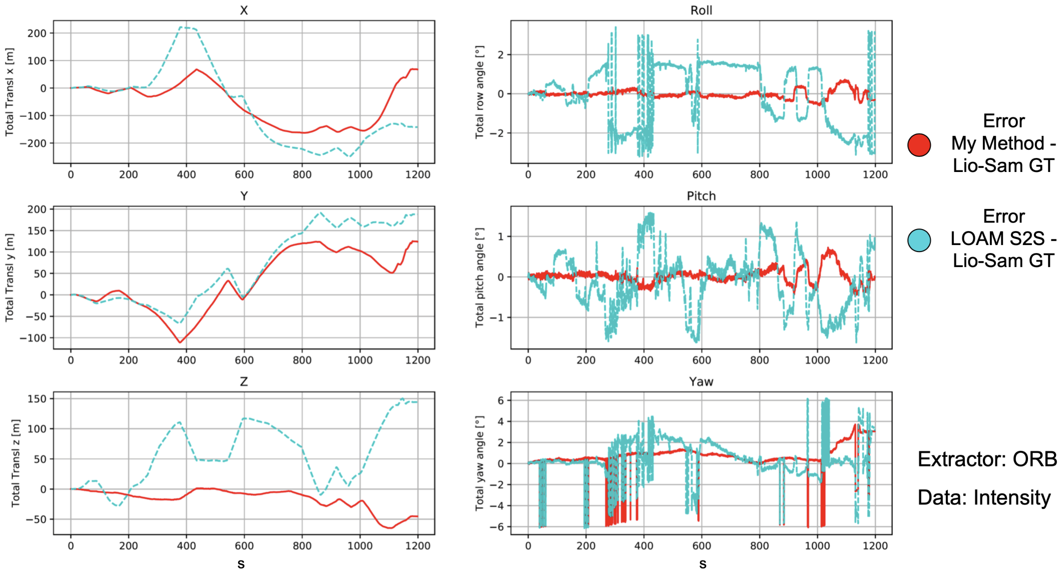

Figure A.3: Pose error comparison orb

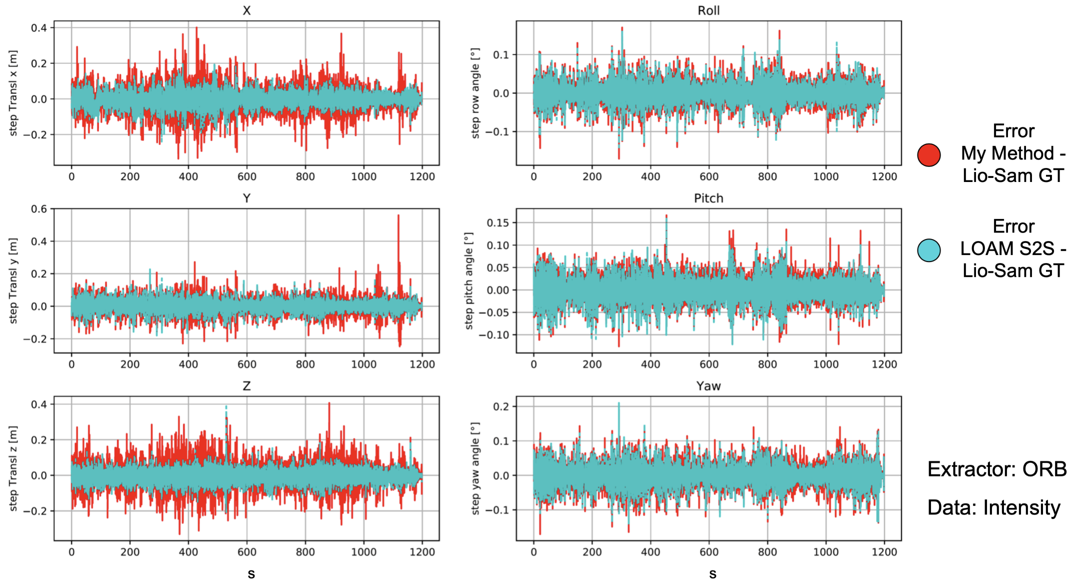

Figure A.4: Step error comparison orb

A.2 BRISK Comparison Plots

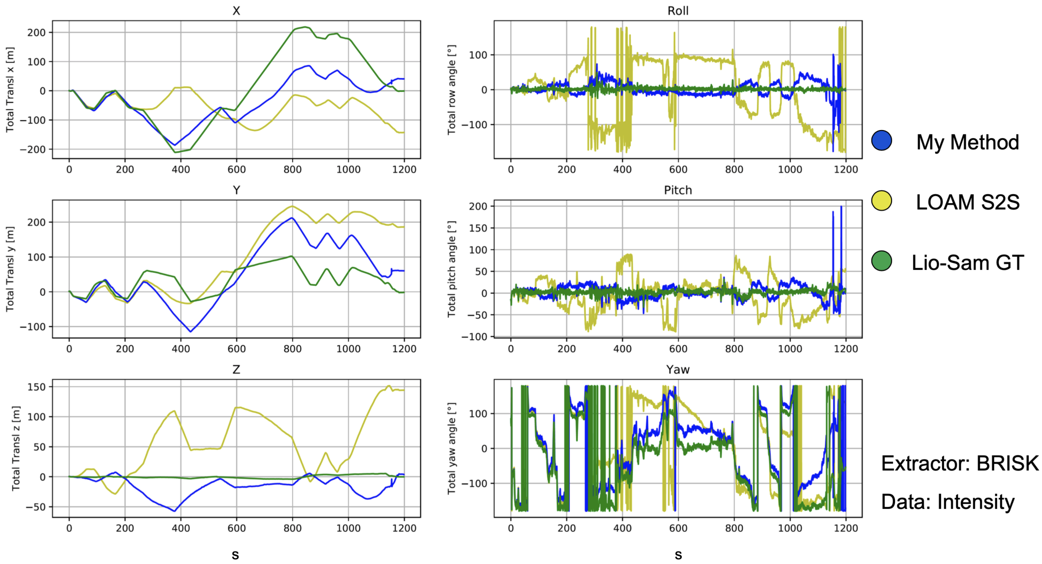

Figure A.5: Pose comparison brisk

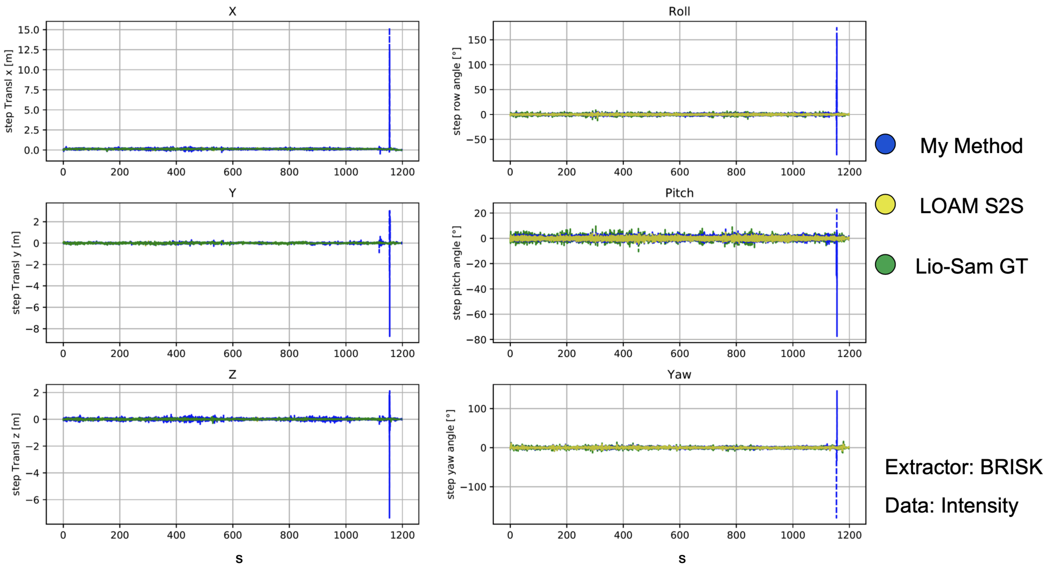

Figure A.6: Step comparison brisk

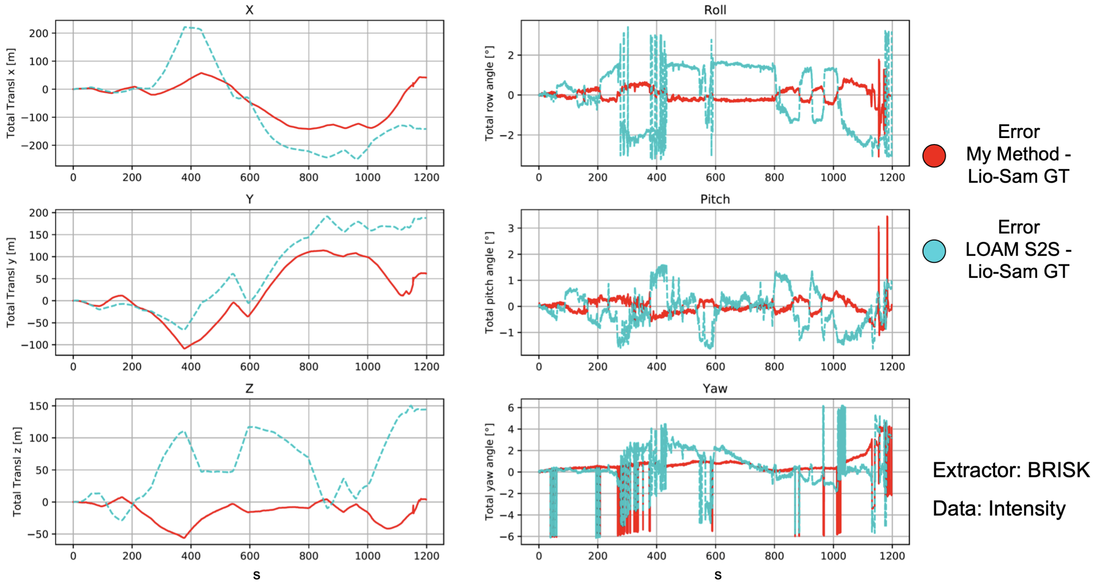

Figure A.7: Pose error comparison brisk

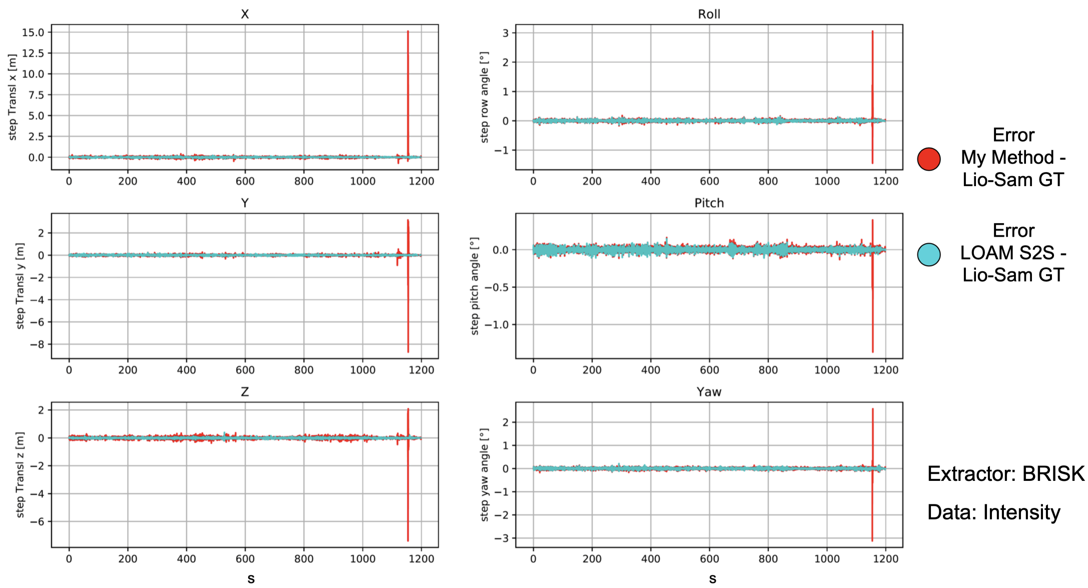

Figure A.8: Step error comparison brisk

A.3 KLT Comparison Plots

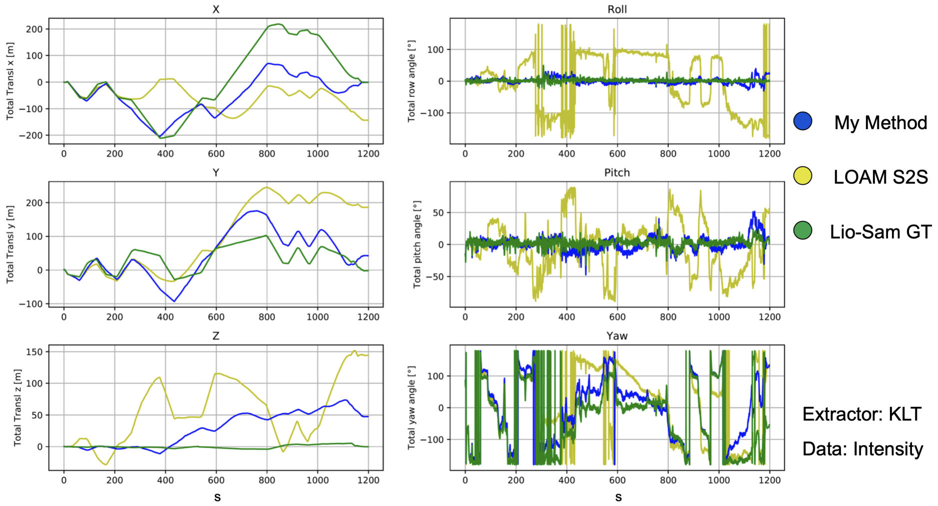

Figure A.9: Pose comparison klt

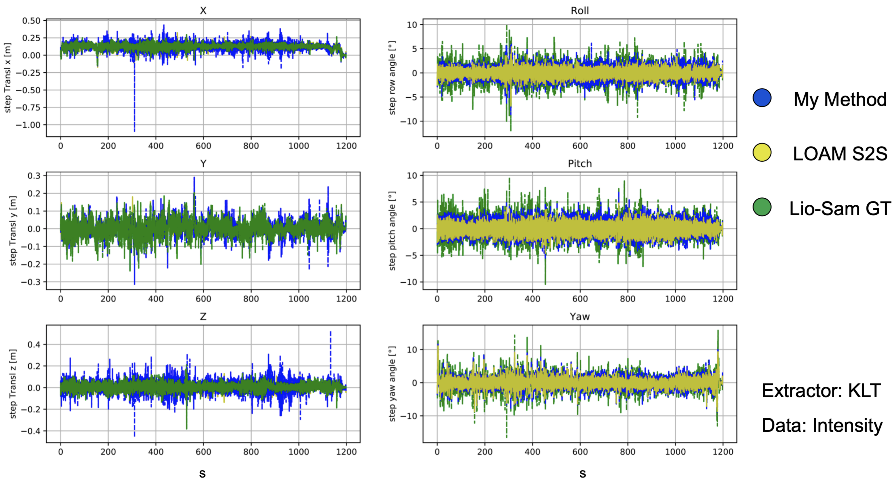

Figure A.10: Step comparison klt

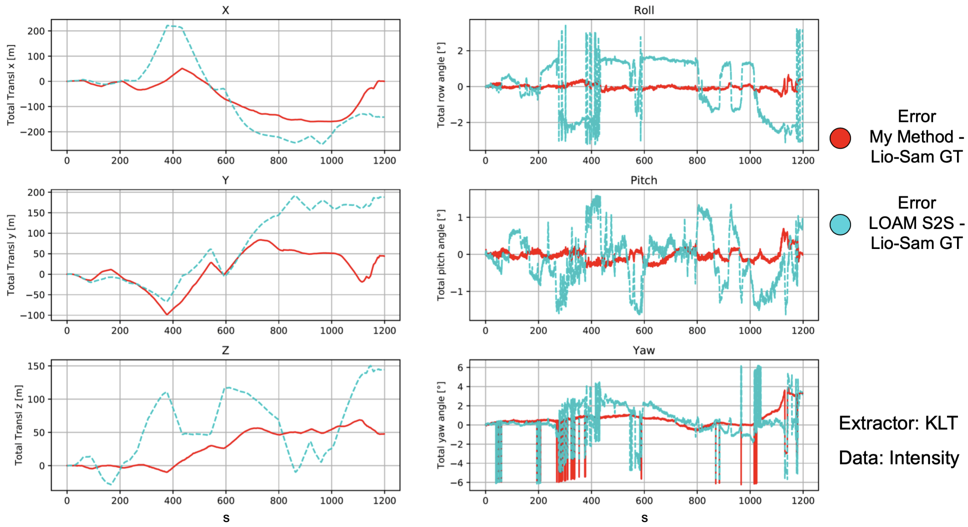

Figure A.11: Pose error comparison klt

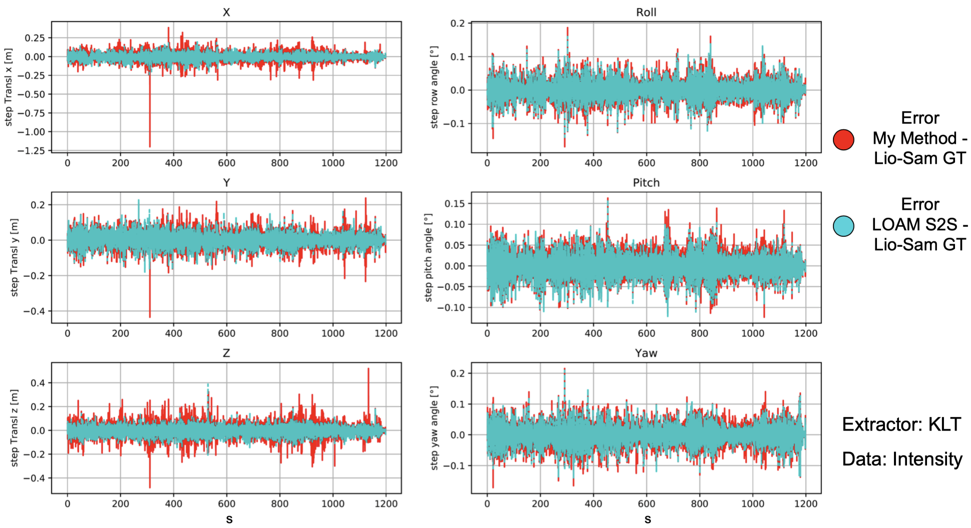

Figure A.12: Step error comparison klt

A.4 Intensity Data Comparison Plots

For the intensity data consider appendix A.1 as the comparison base was intensity and ORB and they are thus the same.

A.5 Ambient Data Comparison Plots

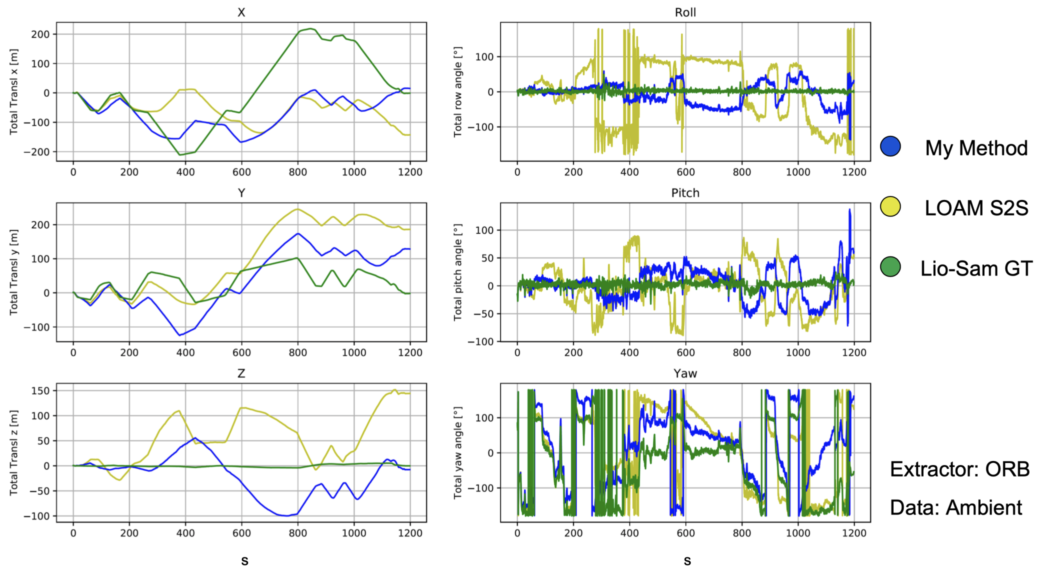

Figure A.13: Pose comparison ambient

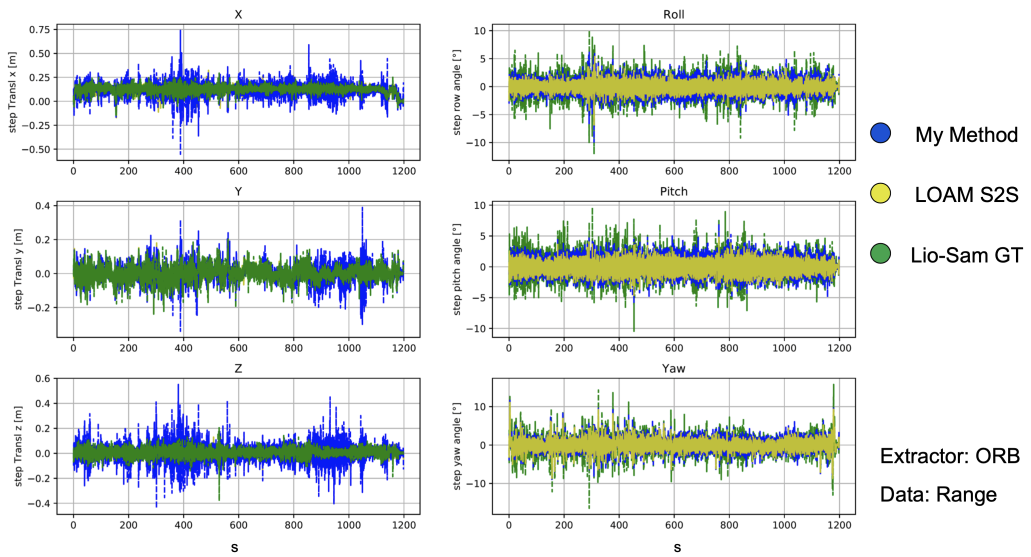

Figure A.14: Step comparison ambient

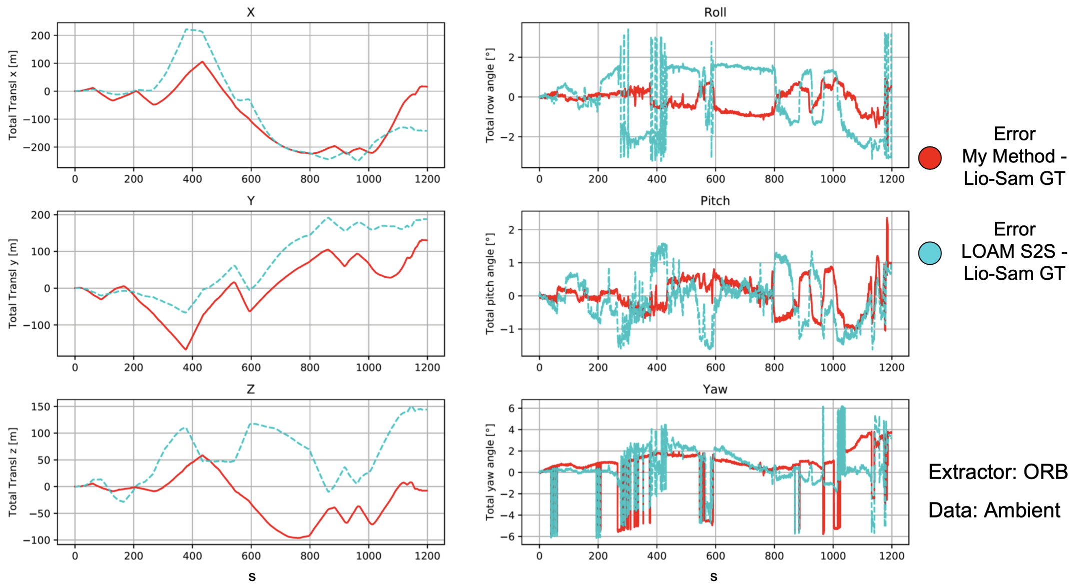

Figure A.15: Pose error comparison ambient

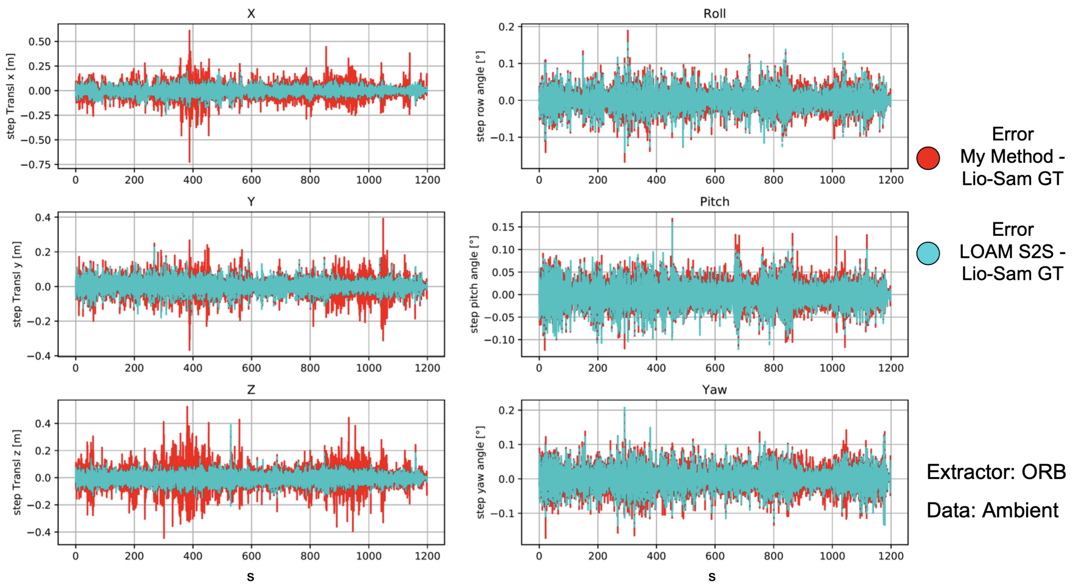

Figure A.16: Step error comparison ambient

A.6 Range Data Comparison Plots

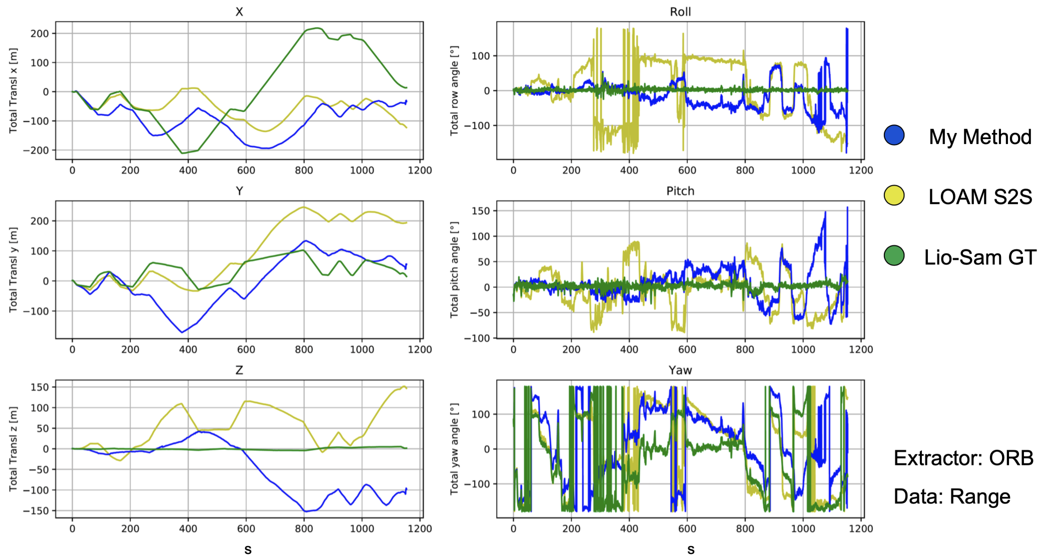

Figure A.17: Pose comparison range

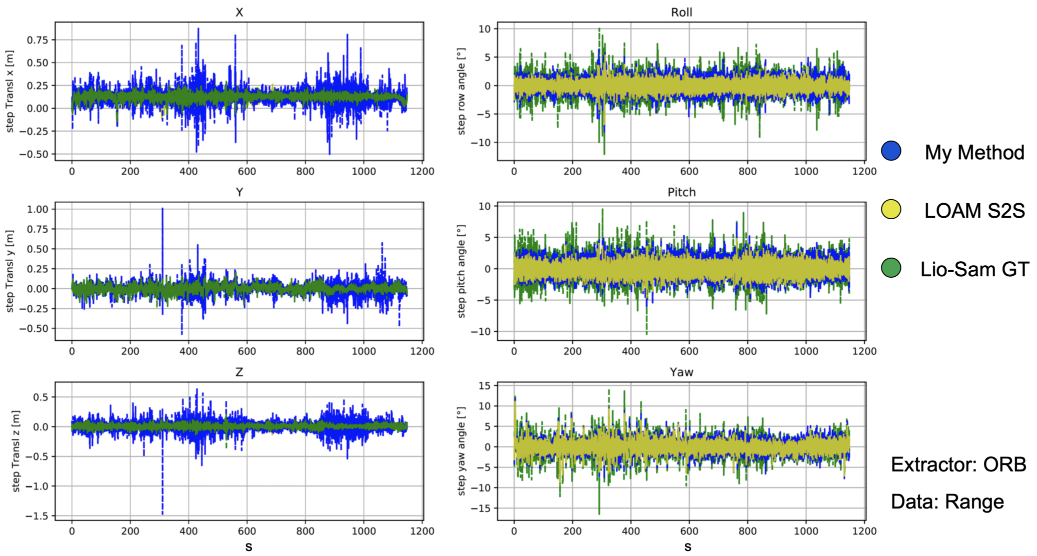

Figure A.18: Step comparison range

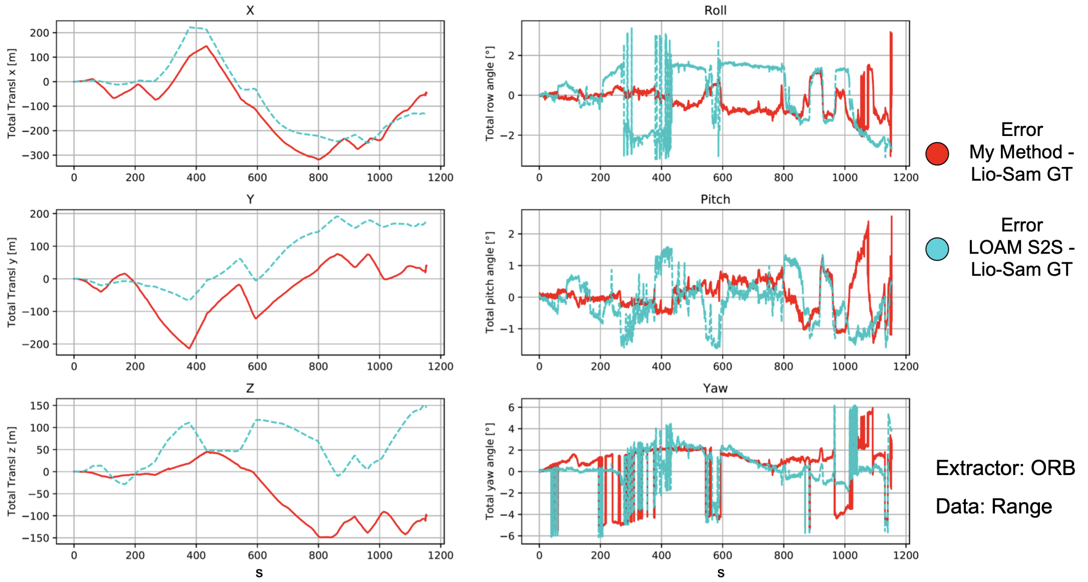

Figure A.19: Pose error comparison range

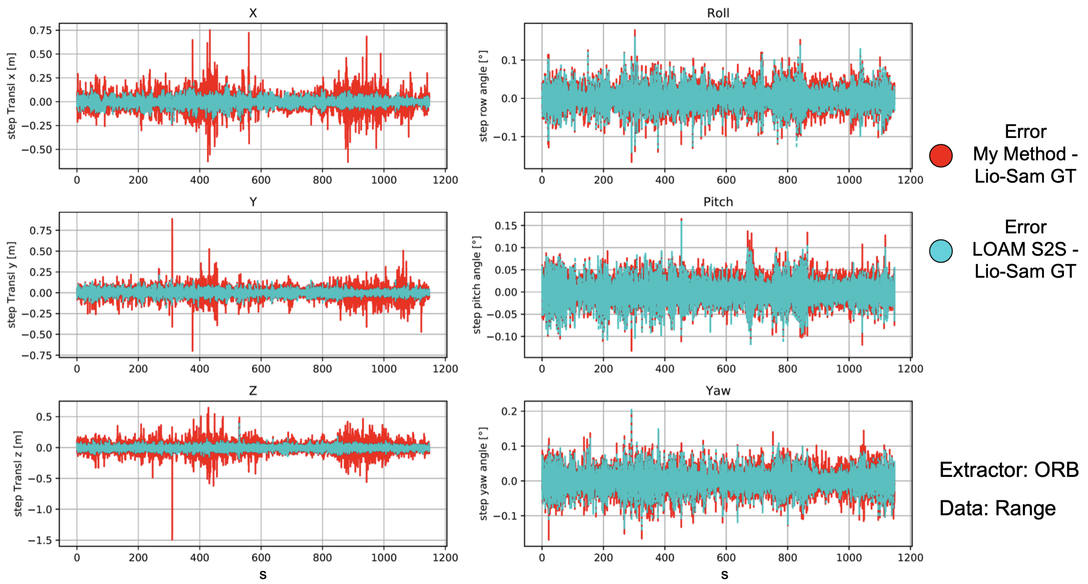

Figure A.20: Step error comparison range