

INFRASTRUCTURE-FREE MAPPING AND LOCALIZATION FOR
TUNNEL-BASED RAIL APPLICATIONS USING 2D LIDAR

by

Tyler Daoust

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Aerospace Science and Engineering
University of Toronto

© Copyright 2017 by Tyler Daoust

Abstract

Infrastructure-Free Mapping and Localization for Tunnel-Based Rail Applications
Using 2D Lidar

Tyler Daoust
Master of Applied Science
Graduate Department of Aerospace Science and Engineering
University of Toronto
2017

This thesis presents an infrastructure-free mapping and localization framework for rail vehicles using only a lidar sensor. The method was designed to handle modern underground tunnels: narrow, parallel, and relatively smooth concrete walls. A sliding-window algorithm was developed to estimate the train's motion, using a Rényi's Quadratic Entropy (RQE)-based point-cloud alignment system.

The method was tested with datasets gathered on a subway train travelling at high speeds, with 75 km of data across 14 runs, simulating 500 km of localization. The system was capable of mapping with an average error of less than 0.6% by distance. It was capable of continuously localizing, relative to the map, to within 10 cm in stations and at crossovers, and 2.3 m in pathological sections of tunnel. This work has the potential to improve train localization in a tunnel, which can be used to increase capacity and for automation purposes.

Acknowledgements

There are several people without whom I could not have completed this thesis. They have supported me through an important transition in my academic and personal life.

I would like to thank Dr. Tim Barfoot for being my adviser and mentor throughout my degree. He saw my potential and pushed me to develop myself academically and personally.

I am very grateful to my parents for providing me with every opportunity that has led me to this point and offering their support to help me pursue and achieve my dreams.

Tremendous thanks to my partner and love, Megan, for the everyday support and motivation that carried me through my entire degree.

I would also like to thank François Pomerleau for his day-to-day support throughout my degree. From overnight field deployments to code development, François was a source of knowledge and inspiration.

Finally, I would like to thank all of my colleagues in the Autonomous Space Robotics Lab for their friendship and support, you are one of the most talented and helpful groups of people I have ever met.

Tyler Daoust

Contents

Acknowledgements	iii
Table of Contents	iv
List of Tables	vi
List of Figures	vii
Notation	ix
Acronyms	x
1 Introduction	1
1.1 Infrastructure-free Localization	1
1.2 Contributions	3
1.3 High-level Overview	4
2 Concept Development	5
2.1 Literature Review	5
2.1.1 Train Localization	5
2.1.2 Mapping and Localization	10
2.1.3 Other Selected Topics	14
2.2 Sensor Selection	15
3 Scan Matching	18
3.1 Iterative Closest Point	19
3.1.1 Point-to-point ICP	20
3.1.2 Point-to-line ICP	21
3.2 Rényi's Quadratic Entropy	22
3.3 Comparison of Methods	25

3.4	Summary	28
4	Methodology	30
4.1	Common Processes	31
4.1.1	Scan Pre-processing	31
4.1.2	Trajectory Smoothing	33
4.2	Map Generation	36
4.3	Localization	37
4.4	Summary	38
5	Dataset Collection	40
5.1	Tunnel Environment	40
5.2	Experimental Setup	43
5.3	Datasets	44
5.4	Evaluation Method and Ground Truth	44
5.5	Summary	47
6	Experimental Results	48
6.1	Mapping	48
6.2	Localization	52
6.3	Summary	57
7	Discussion and Future Work	58
8	Conclusion	62
	Bibliography	64

List of Tables

5.1	October 26th, 2014 Dataset Information	46
5.2	November 30th, 2014 Dataset Information	46
6.1	Localization results for the short range sensor datasets	52
6.2	Localization results for the long range sensor datasets	53
6.3	Localization results using the short range datasets for mapping	54
6.4	Localization results using the long range datasets for mapping	55

List of Figures

1.1	Image of a lidar sensor mounted on the front of a subway train	2
1.2	Image of a typical underground tunnel connecting two subway stations.	3
2.1	Example of an existing train localization system	8
2.2	SeqSLAM image matching diagram	13
2.3	Sensor trade-off study table	17
3.1	Scan matching methods at a glance	19
3.2	Point-to-point ICP example	21
3.3	Point-to-line ICP example	22
3.4	RQE scan matching example	25
3.5	Leica total station used for ground truth	26
3.6	Comparison of scan matching methods without wall removal	27
3.7	Comparison of scan matching methods with wall removal	28
4.1	System overview diagram	30
4.2	Results of wall removal using RANSAC on a typical scan	32
4.3	Data remaining in all scans for run 3 after wall removal	33
4.4	Data remaining in all scans for run 7 after wall removal	34
4.5	Visual representation of the sliding-window filter	35
4.6	Visual representation of the relative localization algorithm	37
5.1	Map of the subway tunnel on which datasets were collected	41
5.2	Examples of the tunnel environments shown using sections of the generated maps	42
5.3	Sensor head used to collect the datasets in the subway tunnel	44
5.4	Sensor mounting location and orientation	45
5.5	Image of a signal post used as a reference marker	47

6.1	Box plots demonstrating the accuracy of all long range maps from the mapping phase	49
6.2	Box plots demonstrating the accuracy of all short range maps from the mapping phase	49
6.3	Trajectory and errors from mapping using the run 7 dataset	50
6.4	Trajectory and errors from mapping using the run 3 dataset	51
6.5	Histogram of the results for the short range datasets	53
6.6	Histogram of the results for the long range datasets	54
6.7	Histogram of the results using the long range datasets for mapping . . .	55
6.8	Histogram of the results using the short range datasets for mapping . . .	56
6.9	Histogram of the of the localization errors for all runs against all maps .	56

Notation

- a : Symbols in this font are real scalars.
- \mathbf{a} : Symbols in this font are real column vectors.
- \mathbf{A} : Symbols in this font are real matrices.
- $\sim \mathcal{N}(\mathbf{a}, \mathbf{B})$: Gaussian distributed with mean vector \mathbf{a} and covariance matrix \mathbf{B} .
- $\mathbf{1}$: The identity matrix.
- $\mathbf{0}$: The zero matrix.

Acronyms

ICP	Iterative Closest Point
RQE	Rényi's Quadratic Entropy
GPS	Global Positioning System
GNSS	Global Navigation Satellite System
RFID	Radio-frequency identification
IMU	Inertial Measurement Unit
RANSAC	RANdom SAmples and Consensus
SLAM	Simultaneous Localization And Mapping
DOF	Degrees of Freedom
KRD	Kernelised Rényi Distance
P2P	Point-to-point
P2L	Point-to-line

Chapter 1

Introduction

1.1 Infrastructure-free Localization

Knowing the location of rail vehicles, such as trains, trams, and subways, is critical to rail-system management, since trains are confined to travel along their railway and have long braking distances. This means that without advanced warning of upcoming obstacles, such as a stopped train, train safety cannot be guaranteed (Bonnett, 2005). To date, all train safety systems rely on track-side infrastructure to determine the location of trains on the railways in that system. This infrastructure is expensive to install, and can be very difficult and costly to maintain. Due to the high costs, track-side infrastructure is interspersed along the railway; the spacing can vary from 10s of metres for modern Radio-frequency identification-based (RFID) systems to 10s of kilometres for conventional commercial transnational railway systems.

All train control systems currently implemented, with the exception of the RFID-based autonomous systems, are categorized as *fixed-block* systems. A block is a defined region of the railway that can only be occupied by a single train. In a fixed-block system, a block is the region between the track-side train detection infrastructure. This system relies on dead-reckoning between track-side infrastructure locations to track the train's

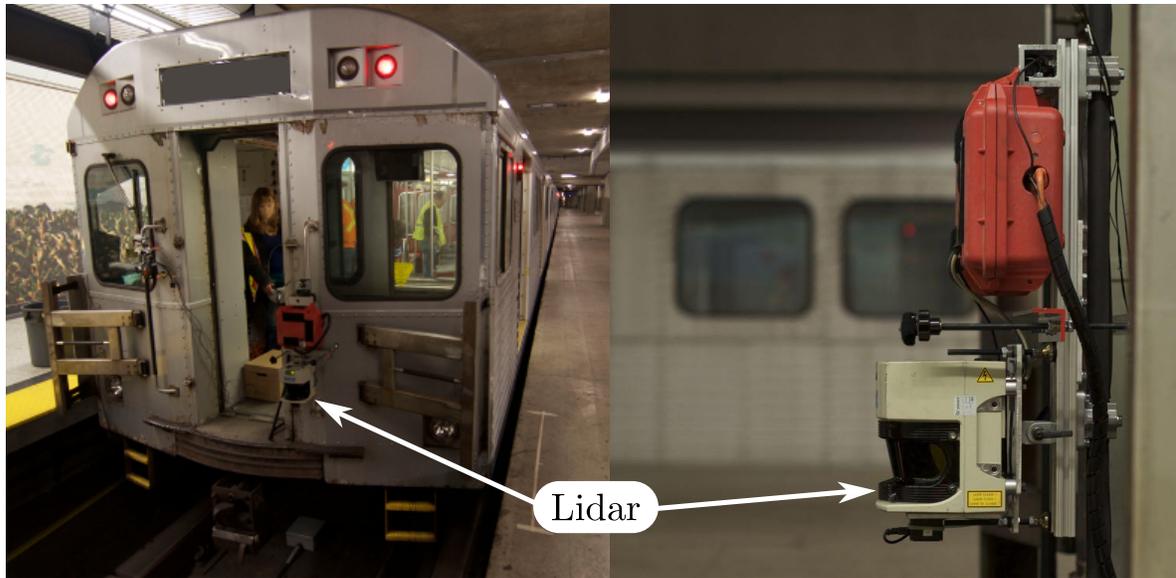


Figure 1.1: A SICK LMS 291 2D lidar unit installed on the front of a subway train for dataset collection. The sensor is positioned such that it is facing away from the train and the 2D scanning plane is parallel to the ground.

position, which is not accurate enough to be used in railway safety systems. A preferred alternative is a *moving-block* system, in which the block is a virtual region that encloses each train in the system, with the size being determined by the speed of the train. The moving-block system allows trains to operate closer together resulting in higher capacities and improved efficiency. However, this method requires continuous updates regarding the position and velocity of every train in the system.

The goal of train localization is to know the precise location of trains to ensure that adequate separation is maintained between them and to provide *track determination*. Track determination consists of identifying on which track a train is travelling when there are two or more parallel tracks or when passing over a switch, a fork, or merger of railways. Thus, the positioning requirement for a train is relative to nearby infrastructure, which can be accomplished using a locally consistent map. Inspired by recent advances in localization systems for high-speed robotics applications (Zlot and Bosse, 2014; Marshall et al., 2008), this thesis presents a proof of concept for the use of a 2D lidar sensor as the sole sensor for estimating the position and velocity of a train. Figure 1.1 shows the

experiment setup used to collect datasets for algorithm development and validation, it includes a lidar unit mounted on the front of a train in an underground tunnel. The use of lidar was chosen because of the nature of the underground railway environment (i.e., dark tunnels) precludes the use of other common sensors, such as cameras and satellite-based system (e.g., Global Positioning System (GPS)). An example of a challenging environment is represented in the image in Figure 1.2, which was captured from a subway train travelling at high speed in a tunnel connecting two stations during dataset collection.



Figure 1.2: Picture of an underground tunnel connecting two subway stations. It highlights some of the pathological conditions for localization that need to be addressed (e.g., low light, smooth wall surfaces and motion blur).

Train position estimation can be viewed as a 1D localization problem as the train is constrained to move on its tracks. This is a simplification of the classic 2D localization problem; however, the speed of the train and the nature of the modern tunnel environment introduce aspects that challenge classical localization techniques.

1.2 Contributions

This thesis introduces an infrastructure-free localization system for railway vehicles capable of a high degree of positioning accuracy to fulfil all localization requirements for rail safety and train operation. Specifically, the novel contributions are as follows:

1. Infrastructure-free mapping algorithm for use in tunnel environments at high speed;
2. Infrastructure-free localization algorithm for high-speed rail vehicles in tunnels; and
3. Experimental validation of both algorithms on a subway train in a real tunnel.

1.3 High-level Overview

The remainder of this thesis is divided into the following chapters. Chapter 2 begins with an overview of train localization in general, progresses into a more detailed review of the sensor sets, and mapping and localization techniques used in robotics, and concludes with a sensor selection trade study that proposes a 2D lidar sensor as the focus for this work. Chapter 3 explores and compares three scan matching techniques used for aligning point clouds obtained from a lidar sensor. Chapter 4 introduces the methodology developed to implement a lidar-based mapping and localization system for use in rail-based tunnel applications. The datasets used in this thesis, as well as the methods used to collect them, are presented and discussed in Chapter 5. Chapter 6 presents and discusses the results of experiments conducted to test the accuracy of the proposed localization and mapping methods using the collected datasets. The final two chapters, Chapters 7 and 8, provide a general discussion of the implementation of the proposed methodology on trains for localization and concluding notes.

Chapter 2

Concept Development

This thesis proposes an algorithm that can localize a train travelling along a previously mapped section of railway tunnel. In order to accomplish this, a literature review was completed, which covers several sensing modalities and localization methods. The primary focus is lidar-based systems, which is based on the sensor selection trade study located at the end of this chapter. The literature review covers signalling and localization techniques, and current autonomous train localization techniques.

2.1 Literature Review

2.1.1 Train Localization

The main purpose of train localization is to ensure that trains do not collide (Bonnett, 2005). Since trains have large braking distances and run on fixed tracks, it is important that trains receive advanced warning about stopped or slow-moving trains in front of them. In the 1860s the first method for detecting a train on a track was developed. The system used an arrangement of electromagnets to determine if a train was occupying a particular portion of track. This was the foundation of the *fixed-block* train-signalling system, where a track is divided into fixed sections called blocks. In this system each

block has infrastructure for determining whether a train is situated in the block and informing trains whether or not an upcoming block is occupied with sufficient warning to ensure the train will not enter an occupied block. Although many technological advances have been made in this area, since this *fixed-block* system is present to some extent in all railways systems used today (Bonnett, 2005).

The drawback with current *fixed-block* systems is that the method used to estimate a train's location is a form of dead reckoning for which the associated error grows unbounded within the block. Research to improve localization accuracy includes the use of one or more of the following sensors: eddy current sensors (Hensel et al., 2011) (Boehringer, 2003), Doppler radar (Acharya et al., 2011), Inertial Measurement Unit (IMU), and optical imaging (Shenton, 2008). However, all of these sensors in combination or alone need to be paired with GPS (Beugin and Marais, 2012) (Boehringer and Geistler, 2006) or track-side markings of various design (Acharya et al., 2011) to reset otherwise unbounded error growth in position estimation. Further detail on these sensors, is provided in the sensor selection section, Section 2.2.

These localization systems represent incremental improvements on existing systems and cannot be used for long durations in safety-critical applications, such as collision avoidance and train automation (Hasberg et al., 2012). Moreover, these systems cannot be deployed without infrastructure (or guaranteed GPS availability) and do not capitalize on the recent advances in robotics and autonomous vehicles (i.e., the ability to process large amounts of data from information rich sensors).

Proposed state-of-the-art systems employ what is known as a *moving-block* system (Shafia et al., 2012). A *moving-block* system can estimate a train's position and create a block around the train allowing the minimum space between trains to be based on the stopping distance of the train instead of the distance between track-side infrastructure. When GPS signalling is not available modern systems rely on, individually or in combination, tachometer (Ernest et al., 2004), IMU (Heirich et al., 2013a), Doppler radar

or eddy current sensors to determine the train's velocity in order to estimate the train's position (Lauer and Stein, 2013) between track-side location markers. See Figure 2.1 for an example of such systems on a train.

The only use of lidar for train localization is related to topological landmark detection, explored by Albrecht et al. (2013). In this case, the lidar sensor is used to identify large structures and environment changes (e.g., exiting a tunnel or entering a station), based on a topological map of these features. This information is then used to verify the localization estimates of other sensors and reset dead-reckoning errors.

2.1.1.1 Sensor fusion

Most proposed self-localizing train positioning systems use multiple sensors to determine a train's velocity, from which the distances travelled can be integrated. The sensors used are commonly a version of an IMU or accelerometer and gyroscope, and tachometer/axle encoder, with recent research focusing on Doppler radar and eddy current sensors for more accurate velocity data: Acharya et al. (2011); Heirich et al. (2013a); Bohringer and Geistler (2006); Mazl and Přeučil (2003); Lauer and Stein (2015). The prevailing trend in localization algorithm is to use either a flavour of Kalman filter or particle filter to estimate pose change between sensor samples (Acharya et al., 2011) (Heirich et al., 2012a). However, this method is not reliable on its own due to the inherent unbounded error growth in distance estimation from velocity integration (Heirich et al., 2012a).

2.1.1.2 Error zeroing

Error zeroing is required periodically to correct the unbounded error growth in the position estimate created using sensor fusion (discussed above). In practice this done by periodically telling the train where it is so that localization estimate are relative to the last updated position. Two position updating methods used are: GPS and track-side markers (Acharya et al., 2011).

GPS (or Global Navigation Satellite System (GNSS)) is widely available and relatively inexpensive to implement; however, trains often travel in areas where limited to no satellite reception is available. This has led to the development of systems that can tolerate GPS blackouts using dead reckoning and track-side markers (Acharya et al., 2011).

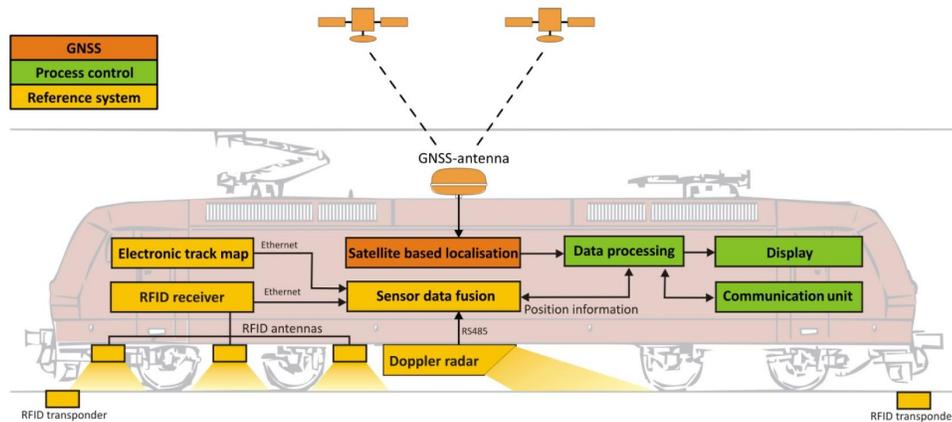


Figure 2.1: Example of a train with both RFID track markers and GNSS for localization, as well as Doppler radar for velocity data. (Manz et al., 2011)

Track-side markers may be passive, active mechanical or electrical devices that inform the train about its current location. The most common marker type is an RFID-like device also known as a *balise* (Acharya et al., 2011). A balise is a stationary device on the track that provides location information when a train passes over it. Each balise on the track is surveyed and programmed to know its location on the track. Thus, trains localize using dead reckoning relative to the last balise over which they have passed. However, the use of track-side markers is expensive to install and maintain, and offers no additional information (Javed et al., 2013) about the track or the environment surrounding the track.

2.1.1.3 Topological localization

A third method for error zeroing, which is discussed in literature but has not yet been implemented in any commercial system, is based on topological mapping. Topological

mapping takes advantage of the known and fixed layout of the railway system. This method uses either IMU-like sensors or eddy current sensors.

The IMU-based method (Heirich et al., 2013a) relies on lateral (perpendicular to the track) acceleration and yaw acceleration to determine the rate of curvature of the current section of track. Presented with an accurate track model, the algorithm can determine when the train has entered and exited a turn, and whether that turn was due to a bend in the track or the train switching to a different track (Heirich et al., 2012a). If the common braking and acceleration points are included in the topological map representation this can provide additional positioning information using longitudinal acceleration.

Another area of interest is adopting a probabilistic approach to localization, based on successes in robotics (Thrun et al., 2001, 2005), and applying it to rail systems. However, the application of these methods for train localization, such as the work by Heirich et al. (2012b), use a topological map and are only used for coarse localization and track determination at switches. The eddy current based topological mapping (Hensel and Hasberg, 2010) creates a map containing the sequence of every switch and joint in the rail system (Geistler and Bohringer, 2004). The algorithm proposed in Hensel and Hasberg (2009) uses a probabilistic approach to determine when a track feature has been passed and what feature it corresponds to on the topological map. The drawback to this system is that the detection and identification of a feature is only completed once the train is meters, sometimes 10s of meters, past a given feature. Another method for velocity and location determination is presented by Heirich et al. (2013b), where the data from a vertically mounted accelerometer is used for vibrations analysis. Heirich et al. (2013b) claim that unique locations and speeds also cause unique vibrations in the train, which can be detected and categorized for future reference. This method is proposed as an addition to other localization systems and not as a standalone system.

Visual techniques are also explored by Wohlfeil (2011) and Corsino Espino et al. (2013). These visual techniques are methods of track determination and are based on

finding switches and determining which track the train is on after passing the switch.

2.1.2 Mapping and Localization

Localization from a known map is the primary objective of the proposed algorithm. This literature review does not focus on Simultaneous Localization And Mapping (SLAM) algorithms because the nature of the application for the algorithm does not require that both mapping and localization occur simultaneously. Most SLAM algorithms rely on a combination of loop closure and estimating pose change between scans using dead reckoning. When a loop closure is detected, the algorithm batch processes many poses to spread any accumulated error throughout the trajectory to minimize local error.

2.1.2.1 Submaps and keypoints

For the purpose of this literature review, scan matching is the process of matching a pair of laser range data scans, or point clouds. In robotics, scan matching usually involves matching orientation, as well as location, to resolve for the 6 Degrees of Freedom (DOF) the robot has while moving in its environment (Nüchter et al., 2010). This topic is covered at great length: Nüchter et al. (2013) and Hahnel et al. (2003). The primary method of matching scans is based on the Iterative Closest Point (ICP) algorithm (Nüchter et al., 2006), which rotates and translates one point cloud relative to another in order to minimize the difference between the two and to determine if there is a transformation that creates a match between the two point clouds. Another method employed is that of 3D occupancy grids usually represented in an Octree form (Nüchter et al., 2007). Another focus of scan matching, which is more relevant to the proposed problem, is dividing the map data into small pieces to enable fast and locally accurate scan matching. The two methods reviewed to accomplish this are creating submaps (Zlot and Bosse, 2009) (Marshall et al., 2008) and identifying keypoints in the scan data (Zlot and Bosse, 2009).

Local submaps can be used to in situations where accurate local positioning is re-

quired and only global consistency is required. The collection of submaps can avoid the compounded errors accumulated while mapping the entire route and to reduce the amount of data contained on an individual map (Marshall et al., 2008). The maps are overlapped such that when the vehicle is transitioning between maps, all of its sensor data lies on both maps, which eliminates the need for loop closure detection (Marshall et al., 2008).

Keypoints are used to group data points that create a significant feature in a map, or a given scan, and create a descriptor for that feature. Feature selection and description of map data is usually done off-line to reduce the processing cost of finding a matching feature from on-line data.

Underground mining applications share similarities to the mapping and localization problems of a train in a tunnel. Zlot and Bosse (2014) used an IMU and a velocity prior for initial pose estimation for a window of scans and then refined the localization estimates by using a sliding-window filter that optimizes the point cloud generated by the lidar sensors. Although they were able to map 17 km of underground mining tunnel, their study was limited to addressing mapping and did not address the accuracy of the map. Mining vehicle localization was addressed by Marshall et al. (2008). Their system used lidar and wheel odometry as inputs to a sigma-point Kalman filter in order to estimate the vehicle's pose and velocity relative to locally consistent successive submaps. While similar to the train tunnel problem, there are several key differences. First, the unstructured tunnel environment in a mine is much more conducive to scan matching than the smooth concrete walls found in a modern underground tunnel. Second, they use wheel odometry as an initial motion estimate between lidar scans.

2.1.2.2 Lidar-only Mapping and Localization

The body of work on lidar-only mapping and localization often relies on feature extraction and ICP for its scan matching: Lingemann et al. (2004); Zlot and Bosse (2009); Zhang

and Singh (2014); Tong et al. (2014). However, the proposed system operates in an environment where there is often not enough data to find features on a per scan basis. Zlot and Bosse (2009) conducted field tests using a car travelling up to 90 km/h and solve a simultaneous localization and mapping (SLAM) problem. However, they do not address the relative localization problem. Lidar intensity data was used by Tong et al. (2014) to form images from 3D scans, which were then processed using visual odometry. However, this requires an expensive and complicated sensor package to form the necessary dense 3D point cloud.

Methods that do not use feature extraction techniques also exist, as explored by Nüchter et al. (2006) and Bosse (2004). However, these methods were not designed to work in the pathological environment that modern subway trains travel: smooth, parallel-walled tunnels.

2.1.2.3 Sequence matching

Sequence matching (Milford and Wyeth, 2012) involves matching sets of data from multiple sequential scans. This method makes the localization much more robust because it is looking for the best matching set of data as opposed to matching individual scans. This technique is beneficial in that speed can be derived from the relationship between the sensor data and the stored data, as seen in Figure 2.2. Although sequence matching is generally used with visual data, Milford (2013) provides examples of extreme image downsampling, which offers the possibility of replacing light intensity with distance information and creating an image matrix that is of size $1 \times N$ (where N is a given number of range data points). This is the same dimension as the data from a 2D lidar sensor.

2.1.2.4 Effects of vehicle speed

It has been noted that scan matching with lidar data decreases in precision as speed increases (Zlot and Bosse, 2009). One reason for the increased error is the assumption

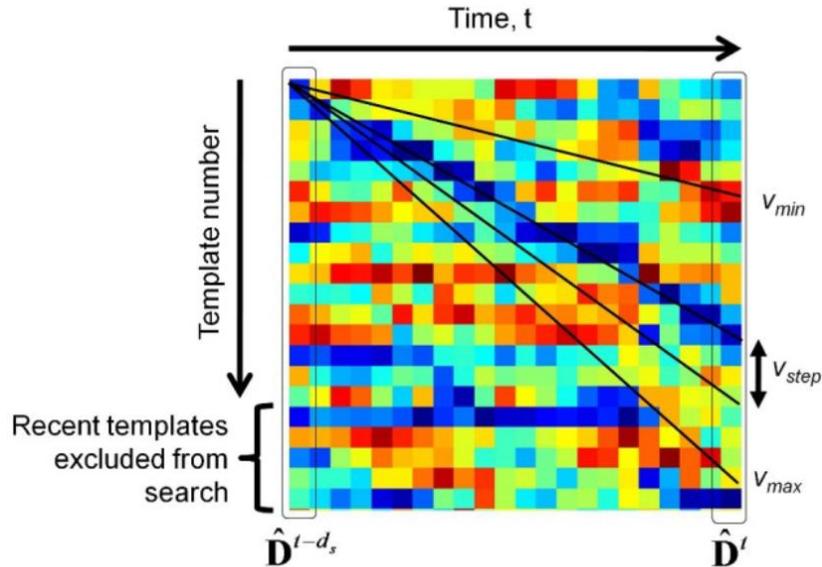


Figure 2.2: SeqSLAM scan matching steps through possible velocities to find best match for the current sequence of images (Milford and Wyeth, 2012) (darker blue squares indicate a strong match).

that a sweep of a 2D or 3D laser range finder occurs instantaneously and from a single location. Motion distortion occurs in individual lidar scans because all data points in a scan (a sweep across the sensor’s field of view consisting of 10s to 1000s or more of individual range measurements) are taken at different times, and therefore, from different places because the sensor is moving. One solution is to stop the robot while scanning for greater accuracy (Tong et al., 2011) (Merali et al., 2012). Another proposed solution is to increase the sensor’s scan rate and speed up localization algorithms (Lingemann et al., 2004). However, this does not solve the issue of motion distortion, it only increases the speed at which the vehicle can operate before its localization algorithm is severely affected. In the case of a high speed train, a single scan could take place over the distance of a meter, which can significantly distort a scan output in relation to a scan of the same area taken while stopped or at a much lower speed. A continuous time algorithm can be used to account for motion distortion in the lidar sensor data cause by variation in speeds. This method tracks the timestamps for individual range measurements and accounts for the distance travelled in the time between measurements (Sheehan et al.,

2013). However, this method requires well-calibrated timestamps, otherwise artificial motion distortion may occur due to timing errors (Elseberg et al., 2013).

2.1.3 Other Selected Topics

2.1.3.1 Localizing along a constrained trajectory

Hasberg et al. (2012) have proposed a probabilistic SLAM technique for vehicles on a constrained 1D curving path, using 3rd order splines to create a probabilistic curve map (PCM) and employing an Extended Kalman Filter (EKF) for state estimation. This is a dead-reckoning-based algorithm, which requires a GPS signal to cope with the unbounded growing position error of the inertial navigation system. Constraining the vehicle to a 1D spline allows for the derivation of vehicle kinematic models to also be performed in 1D. The limitation of the 3rd order spline representation is that it is not an exact approximation of the railway path, which means it is not consistent with the real world track. To improve the path errors, the spline support points are revised every time the train passes; however, approximated paths can never achieve a perfect match to the real world path.

2.1.3.2 Dust modelling

Airborne particles such as dust, smoke, fog or precipitation can affect lidar data. Goodin et al. (2013) describes a method for modelling this interference by characterizing the optical depth of dust clouds. The optical depth is a dimensionless characteristic based on the density and size of the dust cloud. This characterization is important for understanding and accounting for the effect of airborne disturbances when using a lidar sensor outdoors.

2.2 Sensor Selection

The requirements listed in the introduction were used to complete a trade-off study listing possible sensors to be used in the proposed system and their merits, see Figure 2.3. The sensors examined are: GPS, WIFI (signal strength mapping), radar, Sonar (Ultrasonic), 1D, 2D and 3D lidar, and Mono and Stereo Cameras. Whether considered individually or in various groups, an IMU was included in the sensor combination since it was initially thought that an IMU would be necessary in all cases. The result of the study was to select a 2D lidar range scanner with the option of including a GPS input if necessary for outdoor localization. The trade-off study was completed at the outset of the project to refine the scope of the literature review.

The following is a summary of the outcome of the trade-off study. WIFI signal mapping was rejected because of varying signal properties in tunnels with variations in ambient conditions and a lack of external information (to the train) to implement additional features (i.e. obstruction detection). Radar was eliminated because distance estimation from Doppler radar velocity has unbounded error growth, localization from raw radar data is prohibitively computationally expensive, and it is not a proven technology at high speeds (Jose et al., 2010) (Widmann et al., 2000). Sonar was discounted because the sensor's scan rates are too slow: the range is limited and the sensor's readings lack detail for scan matching. Mono and Stereo camera were not selected because of lighting variation concerns (i.e., full dark in a tunnel and varying lighting outdoors). In the end, a 2D lidar was selected because initial research indicated it could meet the design requirements, while 3D lidar was deemed too expensive and 1D lidar can be modelled from 2D lidar data by selecting one or more individual scan angles and discarding the rest. GPS data was tentatively selected for use in the system, as there was concern that some outdoor environments (i.e., on a bridge or in a flat field) may not offer sufficient variable terrain or surfaces against which 2D lidar can localize.

The sensor set and configuration is shown in Figure 1.1. This is an image of the sensor

package mounted on the front of a subway train. A more detailed layout of the sensor mounting package is given in Figure 5.4. Although an IMU was considered to be essential for an end use product and a GPS unit may be required for outdoor localization, this thesis focuses on the sole use of a 2D lidar sensor. This was done to as proof-of-concept to show that such a system can be implemented without an IMU.

Criteria	GPS	Wi-Fi	Radar	Sonar	1D LRF	2D LRF	3D LRF	Single Camera	Stereo Camera
Figures of Merit									
Range - minimum (m)	N/A	0.5	0.25	0.6	0.15	0.5	N/A	N/A	N/A
Range - maximum (m)	N/A	150	200	8	300	250	120	N/A	N/A
Example sensor	 Novatel Smart-AG	 Use existing device	 Continental ARS 300	 SICK UM30-215118	 SICK DL100-23HA2110	 SICK LD-MRS-Heavy Dust	 Velodyne HDL-64E	 Point Grey Blackfly	 Point Grey Bumblebee2
Temperature range (°C)	-40 to 75	Can be in cabin	-40 to 85	-25 to 70	-40 to 75	-40 to 70	-10 to 50	0 to 45	0 to 45
Cost per sensor	\$900	<\$100	\$400	\$400	\$7k	\$20k	\$70k	\$500	\$2k
Distance accuracy	0.5 m	1-3 m	1.5% 0.25 to 3 m	80 mm	3 mm	40 mm	20 mm (0.09 Deg in rotation)	N/A	N/A
Requirements									
1) Map in two passes	Yes / N/A	Yes - assuming signals constant	Yes	Yes	No	Yes	Yes	Yes - scale ambiguity	Yes
2) No additional external infrastructure	Yes - Use existing satellites	Yes - Wi-Fi stations required	Yes	Yes	Yes	Yes	Yes	Yes	Yes
3) 30 cm at <2m/s	No	No	Yes	Yes	Yes	Yes	Yes	Yes, but lighting unreliable	Yes, but lighting unreliable
4) 10m at < 42m/s	Yes	No - variations outdoors	Maybe	No - range too low	Yes - may have trouble outdoors	Yes - may have trouble outdoors	Yes - may have trouble outdoors	Yes, but lighting unreliable	Yes, but lighting unreliable
5) Report at >= 1Hz	Yes	Not Always	Yes	Not always	Yes	Yes	Yes	Yes	Yes
6) Lighting invariant	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No
7) Indoor and outdoor	No	Maybe - signal fluctuations	No	No	No	Yes	Yes	No	Yes
8) Processing requirements	Low	Low	High	High	Medium	High	High	High	High
9) Detect changes/obstructions	No	No	Yes	Yes	Maybe - layout dependent	Yes	Yes	Maybe, but complicated	Yes
10) Cold start localization	Yes (outdoors)	Yes - might be coarse	Maybe	Maybe	No	Yes	Yes	Maybe	Yes

Figure 2.3: Completed trade-off study used to compare proposed sensors.

Chapter 3

Scan Matching

This chapter introduces and compares the scan matching methods that were evaluated for use in the mapping and localization algorithm developed in this thesis. Scan matching is used to align two, or more, point clouds in order to determine the motion of the sensor, in this case a 2D lidar, between scans. This provides two opportunities: estimating the motion of the platform to which the sensor was attached and reconstructing the environment that the sensor traversed through. Both of these functions are used in this thesis. Localization is achieved by estimating the motion of the sensor through the environment when compared to a map of that environment created by combining many previously gathered scans.

Two variants of Iterative Closest Point (ICP) were selected for comparison as well as a third entropy-based scan matching method, Rényi's Quadratic Entropy (RQE). These scan matching methods were selected for comparison because the ICP methods represent two classic scan matching techniques and the RQE method is a relatively new method that offers the possibility of greater accuracy and a smoother objective function.

The selected scan matching techniques are given two point clouds: a reading (moving) point cloud $P = \{\mathbf{p}_1, \dots, \mathbf{p}_M\}$, containing a set of M points, $\mathbf{p}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$, and a reference (fixed) point cloud $Q = \{\mathbf{q}_1, \dots, \mathbf{q}_N\}$, containing a set of N points, $\mathbf{q}_j = \begin{bmatrix} x_j & y_j \end{bmatrix}^T$.

The objective is to determine the distance the sensor travelled between acquiring the two point clouds. Figure 3.1 demonstrates conceptually how an error term for each scan matching method is generated. Each scan matching method is explained and then compared using a dataset collected from an operating subway train to evaluate effectiveness for this application.

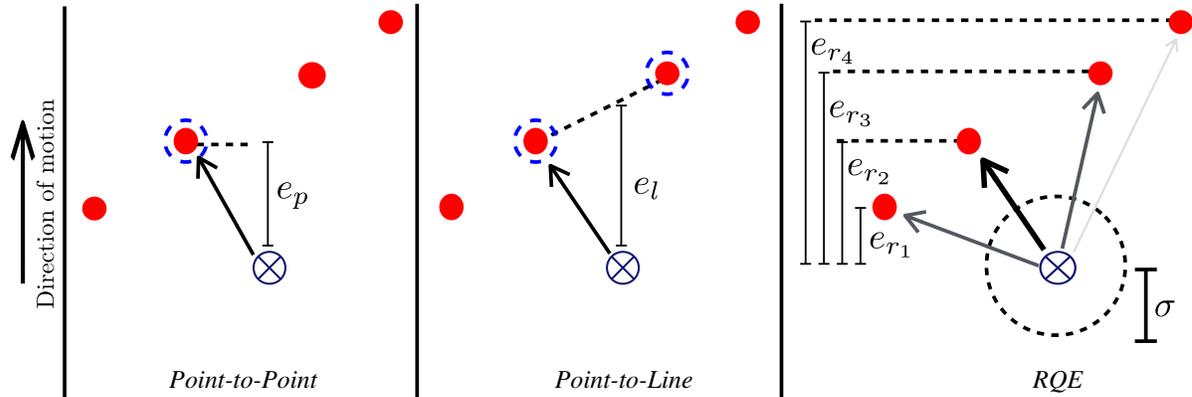


Figure 3.1: General comparison of three scan matching methods. P2P ICP’s error term is based on the distance between each point in the reading, P , scan and its nearest neighbour in the reference scan, Q . P2L ICP’s error term is based on the distance between each point in the reading, P , scan and a line drawn between its two nearest neighbours in the reference scan, Q . RQE’s error term is based on the distances between each point in the reading, P , scan and every point in the reference scan, Q . Due to the constraints of a railed vehicle, only the error parallel to the direction of travel is considered.

3.1 Iterative Closest Point

This section introduces the two variations of ICP scan matching, Point-to-point (P2P) and Point-to-line (P2L), used during the development of the localization and mapping algorithm. ICP is the most commonly used point-cloud alignment technique because it is fast and well understood. ICP finds matches between each point in the moving point cloud to its nearest neighbour in the fixed point cloud. The algorithm then computes an error term based on the distances between the matched point and determines a transform for the moving point cloud that minimizes that error. These steps are then repeated until

the point matches stop changing or other stopping criteria are met. The general form of ICP assumes that paired points between scans are related to the same point on the same surface of the physical object scanned to generate the point clouds. The multitude of variants that have been proposed for ICP attempt to deal with this assumption by trying to infer information about the underlying surfaces of the scanned environment. This often involves interpolating between the points in the fixed scan to generate points on the estimated underlying surface. This is done to provide points to the ICP algorithm that more closely align with location where the points in the reading scan were generated from the scanned surface.

3.1.1 Point-to-point ICP

This is the original and most basic form of ICP. No pre-processing of the fixed scan is done and no interpolated points are generated or used. The P2P ICP error metric is based on work by Besl and McKay (1992):

$$e_p = \sum_{k=1}^N \|\mathbf{p}_k - \mathbf{q}_k\|^2 \quad (3.1)$$

where \mathbf{p} is a point from the moving lidar scan and \mathbf{q} is a point from the fixed lidar scan. The subscript k represents paired points based on a nearest neighbour search.

As eluded to in Figure 3.1, the error metric is modified slightly to account for the constrained 1D motion of the trains, only the component along the track of the error is used. Figure 3.2 shows an example of two simulated scans aligned using P2P ICP. Simulated scans were used to provide ground truth; however, the scans are based on actual scans obtained from a field experiment involving a 2D lidar sensor mounted on a train travelling in a subway. The figure compares the accuracy and shape of the objective function with and without the wall parallel to the direction of travel. Given that the motion of travel is in the positive x direction (left to right), the need for parallel wall

removal to eliminate matches based on scanning pattern becomes fairly evident. The full discussion for wall removal from scans is provided in the next chapter.

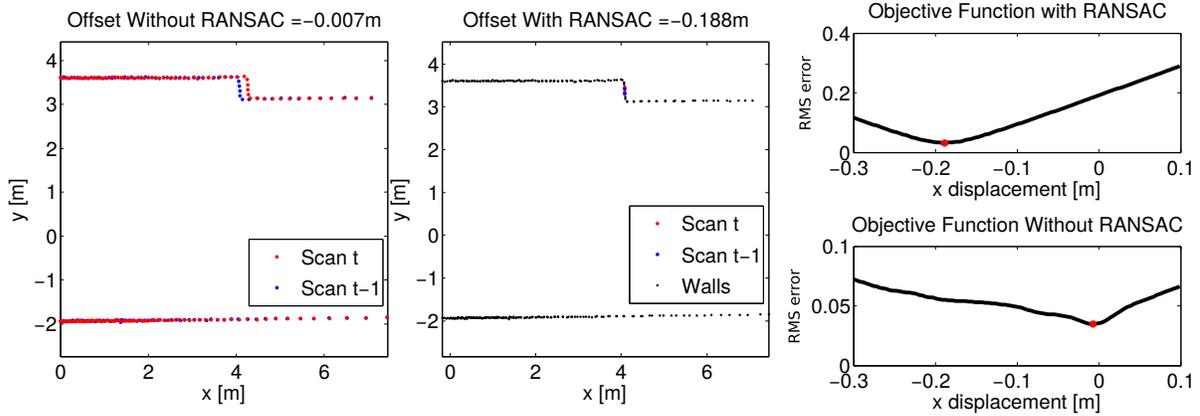


Figure 3.2: Examples of simulated data being aligned using P2P ICP. The offset between the two scans is -0.2 m. The plot on the left shows the optimal alignment of the two scans with the wall data included in the scans. The middle plot does the same but excludes the walls. The right plots are the objective function values for the two alignments.

3.1.2 Point-to-line ICP

The P2L ICP error metric is based on work by Chen and Medioni (1991):

$$e_l = \sum_{k=1}^N \|(\mathbf{p}_k - \mathbf{q}_k) \cdot \mathbf{n}_k\|^2 \quad (3.2)$$

where \mathbf{p}_k is a point from the moving lidar scan and \mathbf{q}_k is a point from the fixed lidar scan. The subscript k represents paired points based on a nearest neighbour search and \mathbf{n}_k is the normal of the line segment approximating the scanned surface near \mathbf{q}_k . This line segment can be calculated in many ways, the most common being a line between the two nearest points in Q to \mathbf{p}_k .

This is equivalent to finding the point at the intersection of the line associated with \mathbf{q}_k and its perpendicular extended from \mathbf{p}_k . This was modified for the given rail-based application such that a line parallel to the direction of travel is extended to intersect with

the line associated with \mathbf{q}_k . This was thought to better account for the constrained 1D motion of the train. This is graphically represented in Figure 3.1.

Figure 3.3 shows an example of two simulated scans aligned using P2L ICP. These are the same two scans used in Figure 3.2. P2L ICP is often used to improve the accuracy of scan matching over P2P; however in this case P2P outperforms P2L and this holds true in the larger example used to compare the three scan matching methods. This method does however perform the best without the walls removed, as expected, due to the fact that the flat nature of the wall surface is modelled by this method.

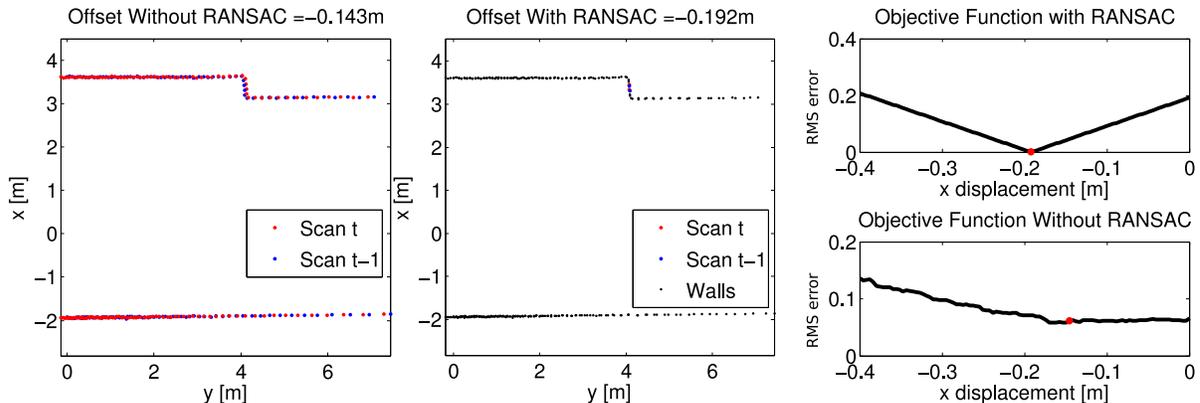


Figure 3.3: Examples of simulated data being aligned using P2L ICP. The offset between the two scans is -0.2 m. The plot on the left shows the optimal alignment of the two scans with the wall data included in the scans. The middle plot does the same but excludes the walls. The right plots are the objective function values for the two alignments.

3.2 Rényi’s Quadratic Entropy

Rényi’s Quadratic Entropy is an entropy-based RQE method as developed by Sheehan et al. (2011); Maddern et al. (2012). In general, RQE computes the distance between each point in a point cloud and every other point in the point cloud, those distances are weighted using a zero mean Gaussian and the results are summed together.

When using RQE to optimally align two point clouds, the Kernelised Rényi Distance (KRD) function is used to assign a measure of entropy to the two point clouds. The

entropy value varies depending on the alignment of the two point clouds and reaches a minimum when they are optimally aligned. Due to the 1D motion constraint, Q is constrained to translation in 1D and x is defined as the alignment offset between the origins of P and Q .

The derivation of the KRD function from RQE is based on the work of Sheehan et al. (2013). The general form of RQE calculates the entropy of all points in a single point cloud, \mathbf{X} of size N , as follows,

$$H_{RQE}[\mathbf{X}] = -\log \left(\frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N G(\mathbf{x}_i - \mathbf{x}_j, \Sigma_i + \Sigma_j + 2\sigma^2 \mathbf{1}) \right). \quad (3.3)$$

Using the general entropy form, Sheehan et al. (2013) derive the following error term, or KRD function, for matching two point clouds:

$$e_r = -\sum_{i=1}^N \sum_{j=1}^M \exp \left(\frac{(\hat{\mathbf{p}}_i - \mathbf{q}_j)^T (\hat{\mathbf{p}}_i - \mathbf{q}_j)}{4\sigma^2} \right). \quad (3.4)$$

This is modified from their cost function to fit with 1D train framework,

$$e(x, P, Q) = -\sum_i^M \sum_j^N \exp(g(x)), \quad (3.5)$$

where

$$g(x) = \frac{-\mathbf{d}_{ij}(x)^T \mathbf{d}_{ij}(x)}{4\sigma^2}. \quad (3.6)$$

The vector distance between the points in the two offset point clouds is defined as:

$$\mathbf{d}_{ij}(x) = \mathbf{p}_i - \mathbf{q}_j + \mathbf{b}x, \quad (3.7)$$

where $\mathbf{b} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ enforces the 1D motion constraint. The isotropic variance, σ^2 , is composed of the noise in the lidar measurements as well as the uncertainty in the distribution of lidar measurements, in the scanned environment. The appropriate value for

σ was determined experimentally and can be used as a tuning parameter to change the convergence of (3.5). A very small σ value results in ICP-like behaviour with a jagged objective function containing many local minima. A large σ value favours aligning the centroid of the two point clouds and results in a flat objective function.

Entropy is lowest in (3.5) when P and Q are well aligned. This means the optimal offset between the two scans can be expressed as

$$x^* = \arg \min_x e(x, P, Q). \quad (3.8)$$

Minimizing (3.8) is done using the gradient-descent method,

$$x_{\text{op}} \leftarrow x_{\text{op}} - \delta \left. \frac{\partial e}{\partial x} \right|_{x_{\text{op}}}, \quad \delta > 0, \quad (3.9)$$

which uses the derivative of (3.5),

$$\left. \frac{\partial e}{\partial x} \right|_{x_{\text{op}}} = - \sum_i^M \sum_j^N \exp(g_{\text{op}}) \left. \frac{\partial g}{\partial x} \right|_{x_{\text{op}}}, \quad (3.10)$$

where

$$\left. \frac{\partial g}{\partial x} \right|_{x_{\text{op}}} = - \frac{\mathbf{b}^T \mathbf{d}_{ij}(x_{\text{op}})}{2\sigma^2}. \quad (3.11)$$

The operating point, x_{op} , is updated until Δx_{op} falls below a predetermined threshold and the minimum is found.

Figure 3.4 shows an example of two simulated scans aligned using RQE. These are the same two scans used to demonstrate both ICP scan matching methods. This method appears to be the most accurate of the selected methods.

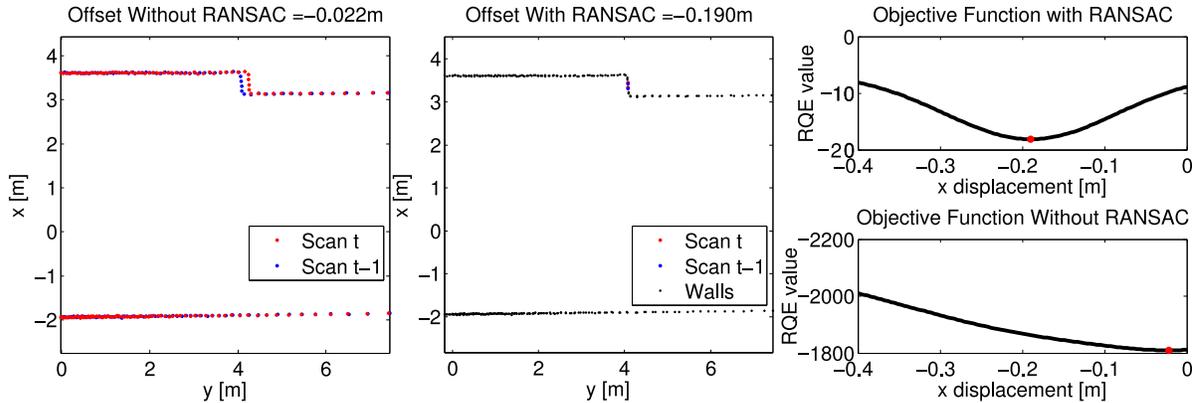


Figure 3.4: Examples of simulated data being aligned using RQE. The offset between the two scans is -0.2 m. The plot on the left shows the optimal alignment of the two scans with the wall data included in the scans. The middle plot does the same but excludes the walls. The right plots are the objective function values for the two alignments.

3.3 Comparison of Methods

The improved performance found using RQE is due to the fact that a lidar sensor only collects a finite number of sample points along the surfaces in its environment and the sample distribution is uneven. These effects are not accounted for in the generic ICP point-matching algorithm and are only partially accounted for in the P2L ICP algorithm.

In order to allow for a better comparison of the presented scan matching methods, a dataset was collected that included very accurate and high frequency ground truth. This was done using the Leica TDRA6000 robotic total station shown in Figure 3.5. The Leica total station uses a 1D lidar sensor mounted on an automated pan-tilt unit, which is capable of tracking a special prism, as highlighted in Figure 3.5. The prism was mounted on the sensor head, that was mounted on the train, providing position ground truth at 10 Hz accurate to a few millimetres. The Leica total station relies on maintaining line of sight with the prism and was limited to about 450 m in the tunnel environment.

A comparison of the three scan matching methods was done by estimating the distance travelled between scans and then feeding that estimate into a basic Kalman Filter to propagate the solution through areas that generate empty point clouds (due to nothing

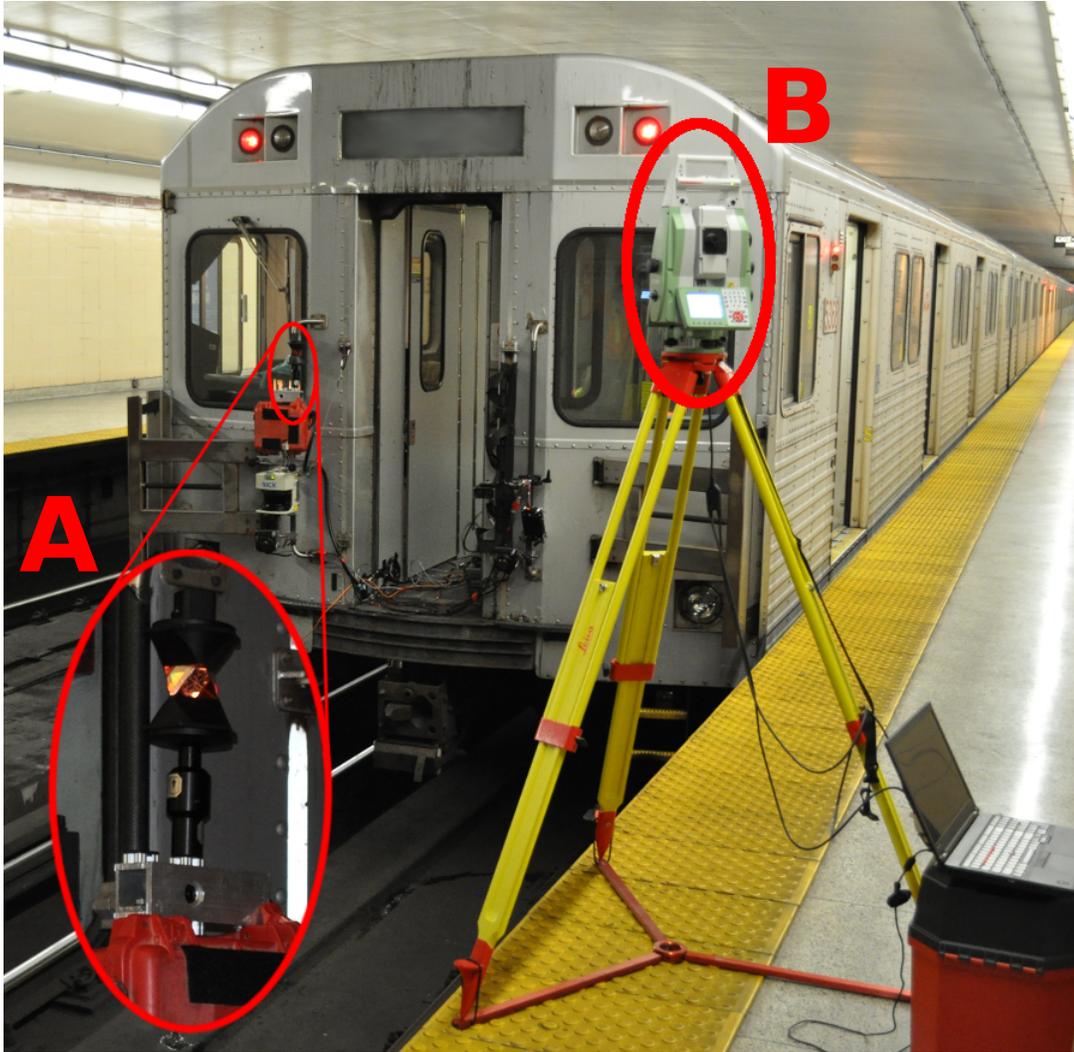


Figure 3.5: The Leica total station (A) is seen on the right side of the image. It is mounted on a station platform such that it maximizes the distance over which the train stays within line-of-sight. Highlighted is the prism (B) that is mounted on the train, this prism is used by the total station to track and measure the position of train as it is moving down the tunnel.

within sensor range or due to parallel wall removal) and to help eliminate some of the noise and error associated with the scans and the scan matching results.

Figure 3.6 shows the trajectory estimated by each of the scan matching methods as well as the ground truth. It can be seen that the P2L method encounters an error from which it cannot recover. The P2P method performs the best in this test; however, it has an error rate near 10%.

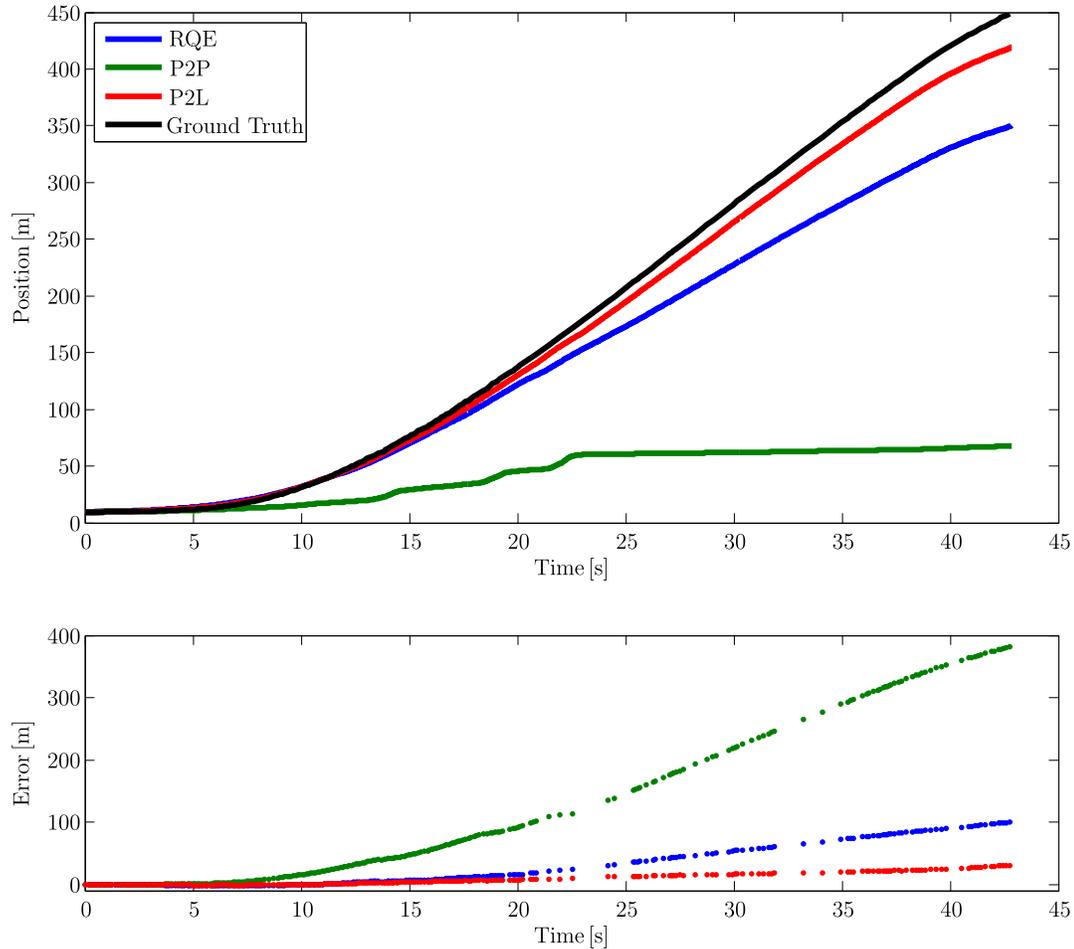


Figure 3.6: Comparison of the three scan matching techniques using the ground truth dataset. The scans have not been preprocessed. Top: Estimated trajectory of the train for each technique as well as the ground truth. Bottom: Estimated error at times that the ground truth was available.

Figure 3.7 shows the trajectory estimated by the three scan matching method when the walls parallel to the direction of travel were removed. This was found to be necessary, especially in areas where the environment was primarily made up of tunnel walls as these walls provide no information that can be used to determine motion along the railway. It can be seen that without the walls, the absolute error and error rate drops significantly for all three methods. RQE has the lowest error, which is very near zero for the section of tunnel tested.

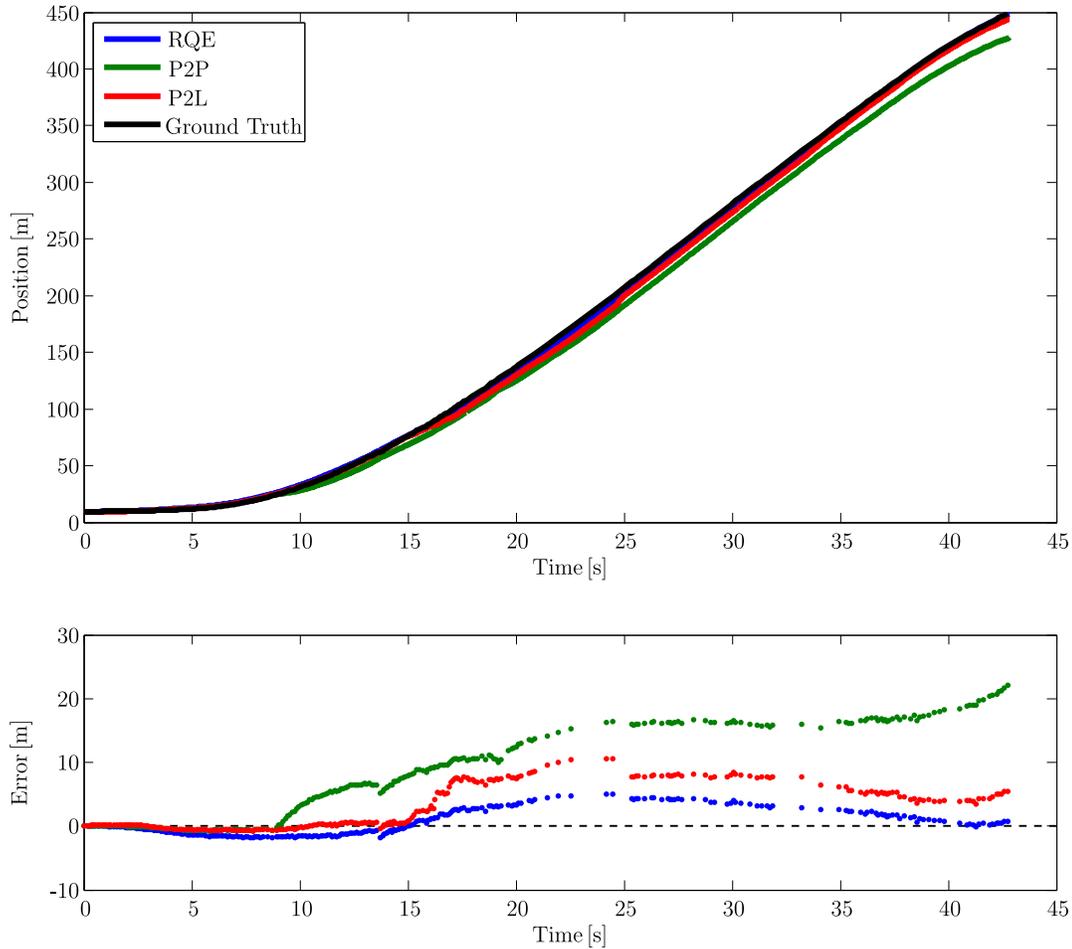


Figure 3.7: Comparison of the three scan matching techniques using the ground truth dataset. The scans have been preprocessed to remove the walls. Top: Estimated trajectory of the train for each technique as well as the ground truth. Bottom: Estimated error at times that the ground truth was available.

3.4 Summary

In this chapter, three methods of scan matching were presented and compared using both simulated data and a dataset gathered from a typical environment in which the algorithm will be used. In all cases P2P ICP performed the worst of the three methods. P2P ICP performed the best initially on the comparison dataset achieving an error rate of 10%. However, the RQE method performed best, and best overall, when the parallel walls were removed, achieving an error rate that did not exceed 1% in the comparison example. Additionally, RQE provides a smooth objective function to be minimized because there

are no changing correspondences in the scan matching algorithm as there are in the two ICP methods. This last point was critical in the development of the sliding window filter presented in the next chapter. Thus RQE was chosen as the scan matching method to be used in this thesis.

Chapter 4

Methodology

This chapter describes the mapping-and-localization framework, which uses the scan matching concepts presented in the previous chapter. A diagram of the process is shown in Figure 4.1. The system is composed of two phases: mapping and localization. Mapping occurs off-line, before the system is in service, and localization is used on-line to report the train's location relative to the generated map. Both phases begin by pre-processing

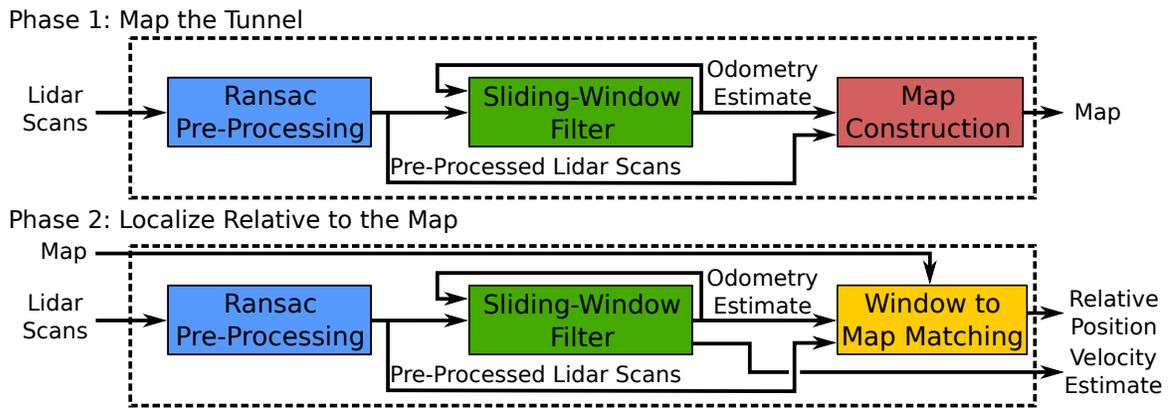


Figure 4.1: System overview showing the flow of data in the localization algorithm. Both phases pre-process incoming lidar scans then calculate the odometry from sequential scans. The Mapping phase generates a map from the odometry, while the localization phase aligns recent scans to a local sub-region of the map. This shows the relationship between the mapping and localization phases. The mapping phase will occur once for a given railway and the localization phase will be used on all subsequent runs on that railway.

the raw lidar data. In both phases, a sliding-window filter is used to find an optimal constant acceleration for a small window of pre-processed scans. In the mapping phase, the estimated positions of the scans are used to register the scans in a global reference frame to create a point cloud, which will serve as the map. In the localization phase, the estimated positions are used to create a point cloud for the window, which is matched to the map to correct the estimated position of the scans relative to the map.

4.1 Common Processes

The first two processes in each phase are identical and are explained in the following section. First, the data is preprocessed, converting the raw lidar data into a usable form and filtering the data to remove uninformative portions. Second, the distance travelled between scans is estimated for use in mapping and localization.

4.1.1 Scan Pre-processing

As described previously, this work focuses on a railway tunnel environment which consists primarily of long flat walls relatively parallel to the rails. During algorithm development, it was noticed that the scan matching techniques that were being explored had difficulty aligning scans. This was because most data in many scans consisted of smooth walls that are parallel to the railway tracks, or to the direction of motion of the train. These parallel walls cause the lidar scans in many sections of the tunnel to appear identical, whether the train is moving or stationary. Thus, it was necessary to remove the data points associated with the walls parallel to the direction of travel, in order to highlight the other surfaces scanned in the tunnels, which actually provide information about the train's motion.

Figures 3.2, 3.3 and 3.4 from the previous chapter show how the objective function of all three scan matching methods become much steeper and more centred on the true

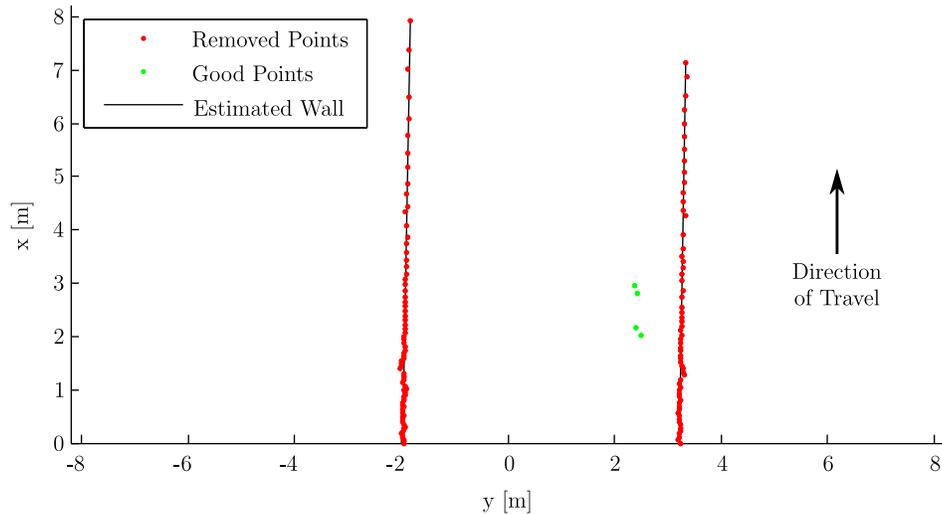


Figure 4.2: This figure shows a nearly featureless tunnel which demonstrated the need to remove the data associated with the walls. Removing the wall data ensures that the scan matching techniques do not superimpose the two scans, as this would suggest that the train did not move between scans.

offset between scans when the parallel walls are removed. RANSAC (Fischler and Bolles, 1981; Zuliani, 2008) is used to identify wall segments within each scan and remove the data points associated with walls. RANSAC was used because it is very fast and robust when used to find lines of points in point clouds. The criteria for classifying a line of points as a wall to be removed was that it needed to be within 7 degrees of parallel to the motion of travel and had to be at least 2 m long. Figure 4.2 shows a scan from a section of tunnel that returns data associated with walls parallel to the direction of travel. The only non-wall data are the points associated with hand rails from a ladder leading from the track. Sparse data such as this are needed in order to localize; therefore the wall data are removed in order to avoid drowning out the useful data (i.e., the data associated with the handrails, protrusions from the walls, or other infrastructure in the tunnels).

Figures 4.3 and 4.4 show the results of pre-processing on a short range and a long range dataset and highlights that minimal useful information is available in the tunnel sections of the railway. The parallel walls are also shown as the grey lines in the point-

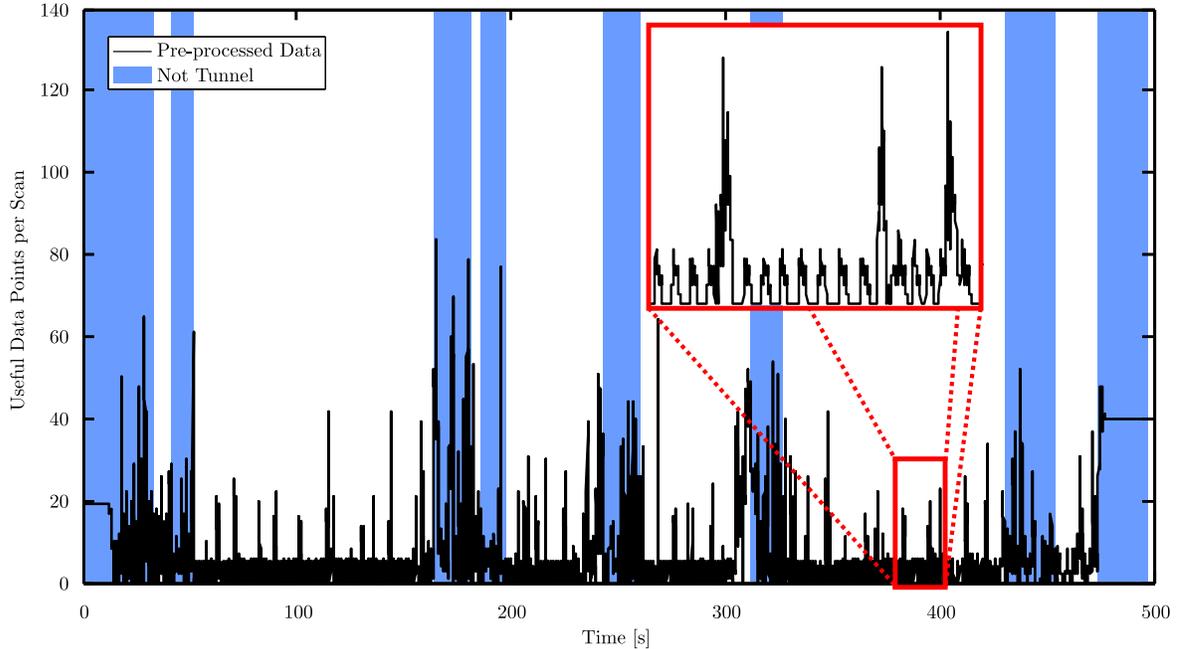


Figure 4.3: The data contained in all the scans for Run 3 are shown after RANSAC processing. Only a small percentage of the 181 data points generated by the sensor are useful, as most return the maximum range value or observe walls parallel to the direction of travel. The highlighted sections of the figure represent sections of the railway that are not tunnel; they are either stations or crossovers. The magnified portion of data shows that there is minimal useful data in tunnel sections and that data gaps frequently occur.

cloud maps shown in Figure 5.2.

4.1.2 Trajectory Smoothing

The sliding-window filter, which provides the odometry estimates, is required for two reasons. First, it provides a smooth velocity profile, eliminating most of the chatter caused by fast scan rates and noise in the scan data. Second, it is used to bridge data gaps arising when the train is passing an area in which the lidar sensor perceives only parallel walls, resulting in empty point clouds after pre-processing.

The size of the window is not fixed in the system due to the nature of the tunnel environment in which the system operates. If the window contained a fixed number of scans, it would be possible to have a window containing no data, as they were all

removed from the scans during pre-processing. Conversely, large windows in stations would result in an excessive amount of data, which would unnecessarily slow the speed of computation. A heuristic was found by selecting the window size based on the number of data points contained within the window, after pre-processing. Another requirement is that the first and last scans in a window contain data. The lower bound for the size of a window is a fixed number of scans in each set. The upper bound is based on a minimum number of data points each set must contain. The second criteria is required because many scans can have little or no data remaining after the RANSAC filter is applied, thus the window will contain as many scans as necessary to include the required minimum number of data points. This ensures that there are both sufficient scans and data in the window to produce an accurate acceleration estimate.

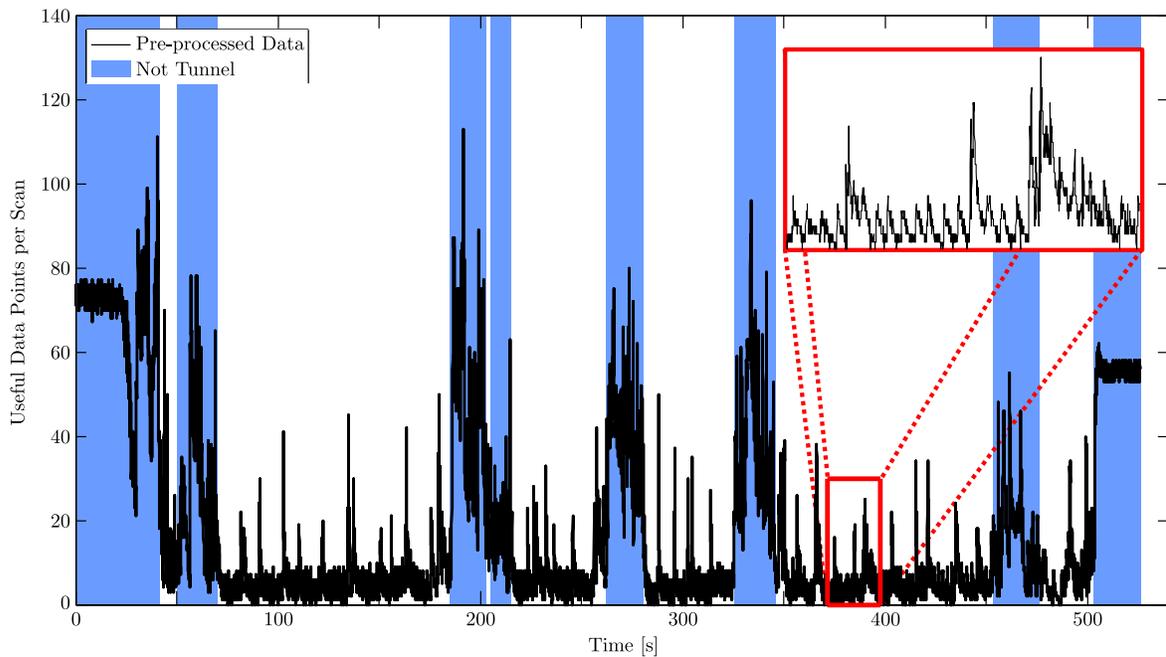


Figure 4.4: The data contained in all the scans for Run 7 are shown after RANSAC processing. Only a small percentage of the 181 data points generated by the sensor are useful. However, there is more data with the longer range sensor setting used in this run than with the previous figure that used the shorter range setting. The highlighted sections of the figure represent sections of the railway that are not tunnel; they are either stations or crossovers. The magnified portion of data shows how little data remain in tunnel sections and how often data gaps occur.

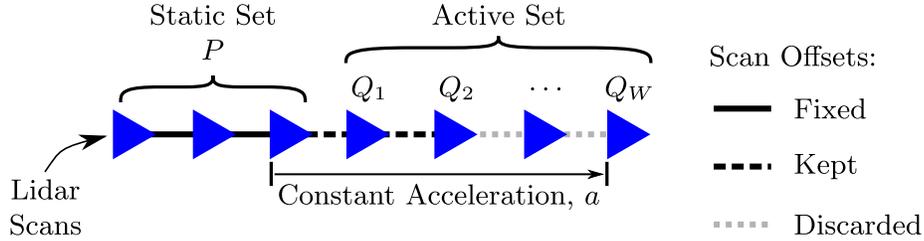


Figure 4.5: Components of the sliding window. The triangles represent individual sequential scans from the localization phase. The sliding window contains scans with known offsets and velocities, the static set, and unknown offsets and velocities, the active set. The static set of scans is combined into a single point cloud, P whereas each scan in the active set remain as individual point clouds, Q_1, \dots, Q_W . The train motion between the scans in the active set is constrained by a constant acceleration, a . Because the sliding window can contain many scans only some of the calculated values from the active set are kept; those that are discarded will be calculated again in future iterations as the window slides forward. The last (rightmost) scan of the static set provides initial velocity, v_0 , for the active set.

The optimization of the sliding window for a constant acceleration over the window is a simplified version of continuous-time trajectory estimation Furgale et al. (2015); Anderson et al. (2015). This work similarly assumes a constant acceleration model to space scans along a trajectory. The sliding window smooths the trajectory by limiting any discrete changes in velocity, which is valid for the system as the train has a large mass and accelerates slowly.

The configuration of the sliding window is shown in Figure 4.5. The scans in the window are divided in two groups, the active set, scans Q_1, \dots, Q_W , and the static set, which are combined into a single point cloud P . The offsets between scans in the active set and the last scan in the static set are defined as

$$x_{k0}(a) = v_0 \Delta t_{k0} + \frac{1}{2} a \Delta t_{k0}^2, \quad (4.1)$$

where v_0 is the initial velocity, a is the acceleration value for the window and Δt_{k0} is the time between the k^{th} active scan and the last scan in the static set. The offset between

scans in the active set is defined as

$$x_{kl}(a) = v_0 \Delta t_{kl} + \frac{1}{2} a (\Delta t_{l0}^2 - \Delta t_{k0}^2), \quad (4.2)$$

where Δt_{kl} is the time between the k^{th} and l^{th} active scans.

The entropy of the window is calculated by the summation of the entropy of each pair of point clouds in the window,

$$f(a, P, Q_1, \dots, Q_W) = \sum_{k=1}^W e(x_{k0}(a), P, Q_k) + \sum_{k=1}^{W-1} \sum_{l=k+1}^W e(x_{kl}(a), Q_k, Q_l), \quad (4.3)$$

and the optimal acceleration,

$$a^* = \arg \min_a f(a, P, Q_1, \dots, Q_W), \quad (4.4)$$

is found by minimizing that entropy.

The gradient descent method, described above, is used to find a^* , this time operating about a_{op} . The derivative of f is found using the chain rule with the derivative of e , above, and $\frac{\partial x}{\partial a}$ giving,

$$\left. \frac{\partial f}{\partial a} \right|_{a_{\text{op}}} = \sum_{k=1}^W \frac{1}{2} \Delta t_{k0}^2 \left. \frac{\partial e}{\partial x_{k0}} \right|_{x_{k0}(a_{\text{op}})} + \sum_{k=1}^{W-1} \sum_{l=k+1}^W \frac{1}{2} (\Delta t_{l0}^2 - \Delta t_{k0}^2) \left. \frac{\partial e}{\partial x_{kl}} \right|_{x_{kl}(a_{\text{op}})}, \quad (4.5)$$

which is used to update a_{op} until the minimum is found.

4.2 Map Generation

The mapping phase generates a static map of the railway environment based on an initial pass of the train mounted lidar system through the railway tunnel environment. This phase is intended to be done off-line to create an accurate and dense point-cloud survey of the tunnel system that will be used for localization.

Using the output from odometry, the position of each scan relative to the first scan can be calculated. The global map is registered by selecting a subset of equally spaced scans, based on their position relative to the first scan, to ensure that the map point cloud is not overly dense. It was empirically found that a spacing of 0.5 m between scan was a good trade off between an overly dense map and losing information. This means that tunnel mapping can occur at the maximum speed of the train as the sensor scans at 75 Hz and the maximum speed of the test train is 70 km/h (or 0.26 m/scan).

4.3 Localization

Localization to the map is done by matching the current window of scans created during the odometry step (which can now be combined into a single point cloud, Q) to a local submap, P , extracted from the map, detailed in Figure 4.6. The local submap is created by selecting all data points in the map that are in an area the size of Q . Extra data points from the map are added on either end to ensure the area represented by the window is wholly contained within the local submap. The location of this region on the map is given by the known relative position of the scans in the static set.

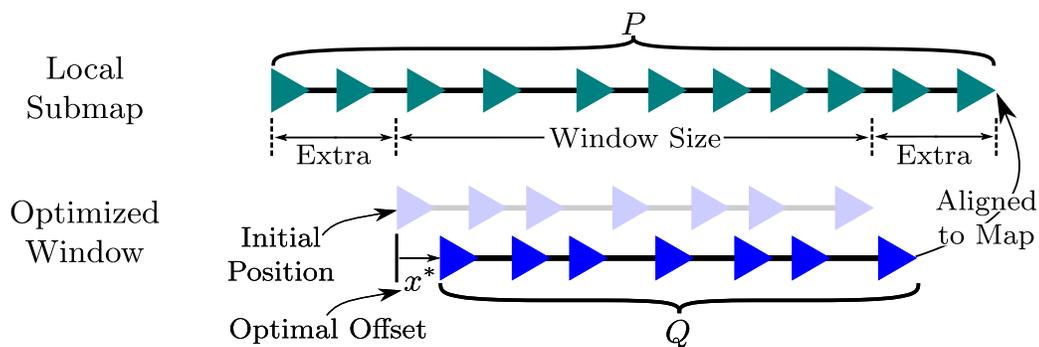


Figure 4.6: Relative localization, illustrating how the sliding window and local submaps are aligned. The triangles represent individual scans from a mapping phase (top) and a localization phase (bottom). The optimal alignment, x^* , of the window point cloud, Q , is determined relative to the local submap, P , using (3.8). The local submap is generated based on the initial position and the size of the window with additional scans included on either end to allow for misalignment.

The relative position of the window with respect to the map is obtained using (3.8). The optimal offset x^* , of the current window, Q , is applied to the individual scan offsets, which are kept from the odometry step to account for any discrepancies in the odometry between the mapping and localization phases. This ensures that the location of each scan is known relative to the origin of the map.

Due to the large number of data points in both Q and P , it was computationally expensive to compute x^* . Therefore, both point clouds were randomly downsampled by 50% to reduce the computation time by 75%. In order to decrease the number of iterations to converge and avoid converging to a local minimum two techniques were employed. First, a global search was performed in an area of ± 1.5 m from the estimated position in the local submap with 0.2 m for a best match to find the global minimum. Then, a technique called the *golden search method* was employed to find the minimum in the area found in the previous step. It is a version of the Fibonacci search devised by Kiefer (1953). The golden search method is one of the fastest ways to find an extremum of unimodal function, although it only requires that the function is unimodal between the two bounds it is given to search. The golden search method iteratively bisects the area between its search bounds using the golden ratio $\varphi = \frac{1+\sqrt{5}}{2}$, determines which section contains the extremum and repeats the process until the search bounds shrink to a predefined size.

4.4 Summary

In this chapter, the methodology for pre-processing scans, map generation and train localization are presented. Both the mapping and localization phases are based on the odometry method developed using a sliding window filter that assumed a constant acceleration over the length of the window of point clouds. The idea of removing point-cloud data associated with surfaces parallel to the direction of travel was fundamental to the

success of this algorithm. Removing the parallel walls from the data greatly increased both the accuracy and the processing speed of the algorithm. The sliding-window filter provides a smooth velocity profile for the estimated trajectory of the train and contributed significantly to the high degree of accuracy reported in Chapter 6.

Chapter 5

Dataset Collection

The experiment environment consisted of a subway train travelling in an underground tunnel. The subway line is a 5.3 km long railway that is relatively straight and completely underground. Multiple datasets were collected, with varying maximum train speed, ranging from 15 to 70 km/h. The layout of the test environment is provided at the top of Figure 5.1.

5.1 Tunnel Environment

The subway line in which the datasets were collected consists of three types of railway environments. Images of these environments are provided in Figure 5.1. Further, maps of these environments (generated by the algorithm) are presented in Figure 5.2.

The most common railway environment type is the single-track tunnel. These are the straight, narrow, and nearly empty sections that connect stations. Occasionally there are handrails or electrical boxes in the tunnel but they are infrequent, small, and repeated at semi-regular intervals. This is the section of the railway that is most difficult for localization.

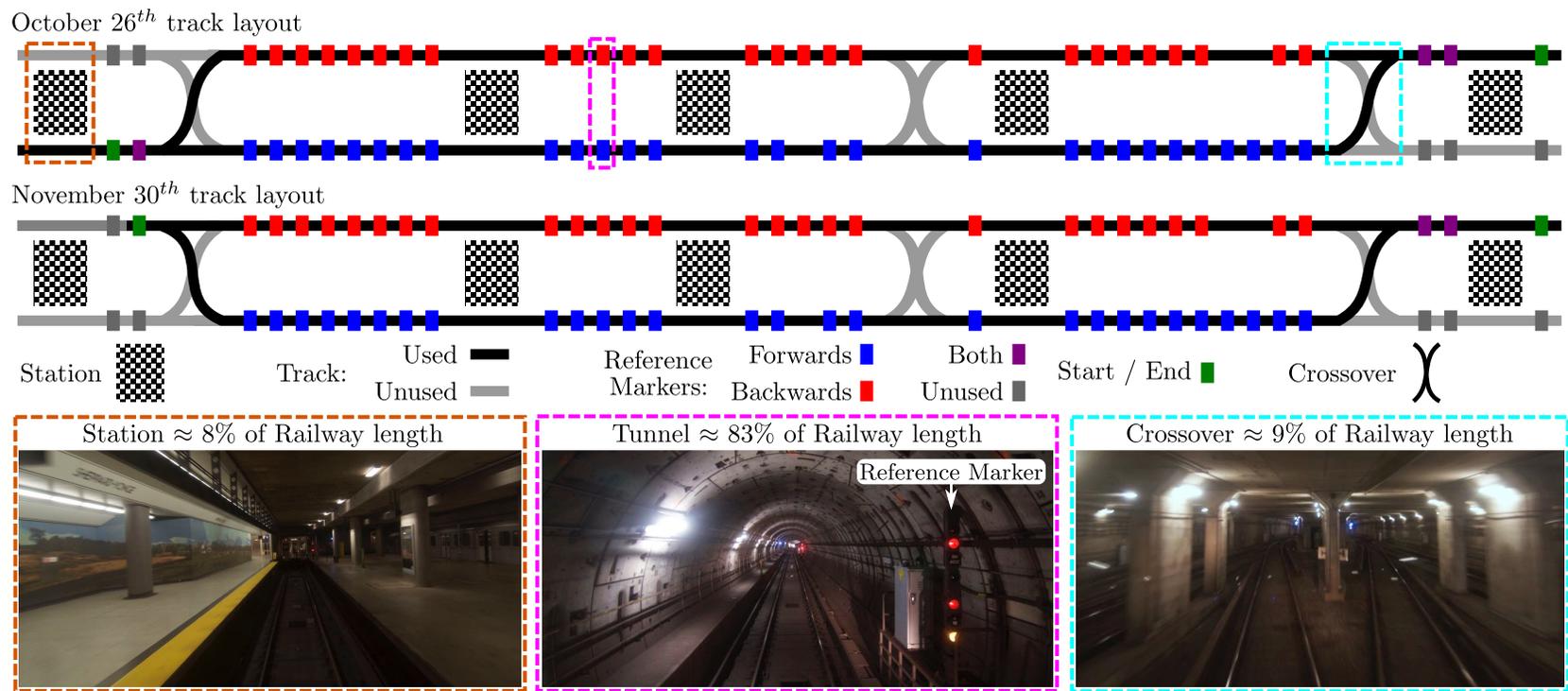


Figure 5.1: Tunnel map of the subway line showing the layout of the test railway as well as typical pictures of the three types of tunnel environments. Both datasets were collected on the same subway line; however, one of the start/end points were different between the two field trials. This difference is shown on the two track layouts for the two field trials. The train travelled forward on the bottom track and backward on the top track and the transfers at the end of each run were used to get back to the start/end points. Green markers denote the first or last marker seen by the train, purple markers were seen by traverses in both directions, red and blue markers are on the top and bottom tracks respectively. Greyed-out reference markers and sections of track were not traversed during these experiments.

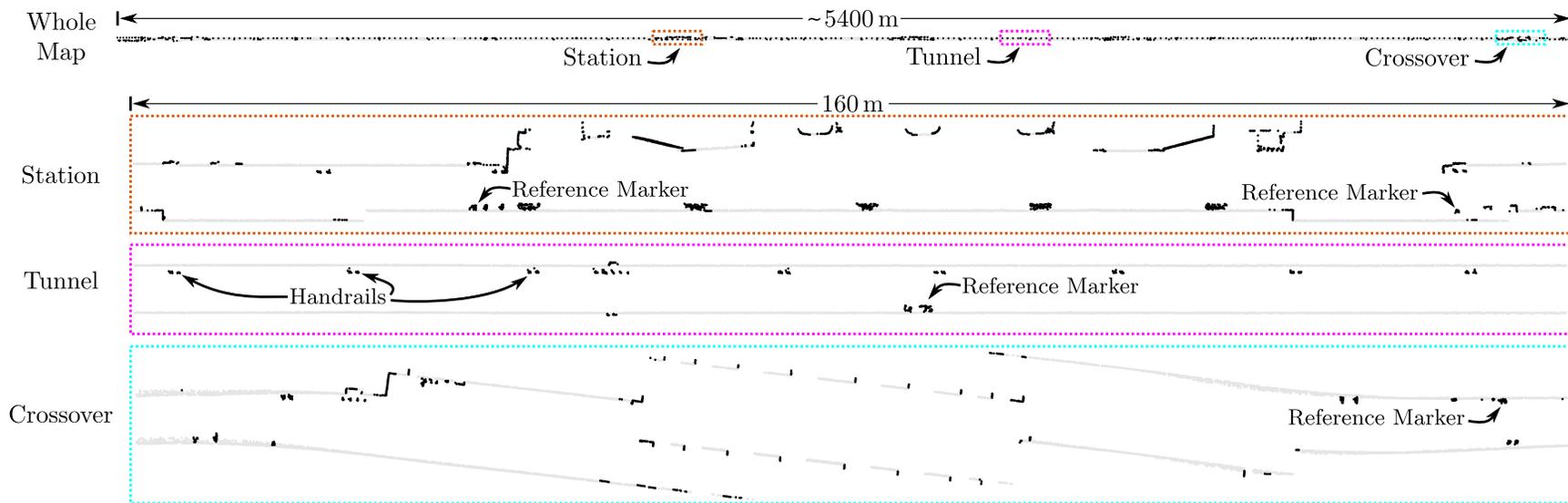


Figure 5.2: Mapping result. The point cloud generated during the mapping phase for Run 3 with enlarged sections of interest. Black data points represent the data after pre-processing and the light grey data points were associated with walls and removed from the data. Examples of the three tunnel environments (station - orange, tunnel - magenta, crossover - cyan) are identified on the map and enlarged below. Each highlighted point cloud consists of hundreds of individual scans and the apparent crispness of the resulting map demonstrates the effectiveness of the odometry estimates. Also identified are the reference markers in each highlighted point cloud and some examples of handrails in the tunnel environment.

The second-most common environment is the station platform. These are characterized by many irregularities such as pillars, stairwells, and large open spaces. Station platforms are often preceded by larger, often irregular tunnel sections, making them easy to identify and providing many perpendicular surfaces that allow for precise localization. Stations also offer many perpendicular surfaces and non-wall structures and are traversed at low speeds, making them ideal for localization. However, stations are also the largest source of dynamic noise relative to the pre-generated maps due to pedestrians and relatively frequent construction.

Finally, the third railway environment is the crossover. This is an area usually just outside a station platform where trains can switch between the two parallel tracks. In the experiments, the trains only turned significantly while using crossovers at the end of each run.

5.2 Experimental Setup

A sensor head, see Figure 5.3, was mounted on one end of the train. It included the lidar sensor, an IMU and a web camera. The lidar sensor head was mounted on the train such that the 2D lidar was facing away from the train, pointed straight forward, so that the scanning plane was parallel to the ground, see Figure 5.4. The SICK LMS291 sensor used to gather the dataset had a scan rate of 75 Hz, an angular resolution of 1 degree adjustable range and accuracy settings. The two common settings are 8.1 m range with millimetre accuracy and 81 m range with centimetre accuracy.

The IMU data was collected in case it would be needed in the empty sections of tunnel between station but was not required and therefore not used in this work. The web camera was not used as an input to the algorithms but as a visual aid during algorithm development, testing and demonstration. It was also used to help find the reference markers discussed later in this chapter.



Figure 5.3: Experimental setup used to gather data in the subway tunnel. The sensor rig consisted of the SICK LMS291 lidar sensor, a microstrain IMU and a webcam.

5.3 Datasets

Table 5.1 describes the six runs from the first dataset and Table 5.2 describes the eight runs from the second dataset. Both datasets were collected on the same subway line; however, the paths followed in the two experiments were not identical, as shown in Figure 5.1. The first dataset was collected using the shorter range and more accurate setting of the lidar sensor with the thought that the higher accuracy would provide more detail about the wall surface than additional information gained farther down the narrow tunnel with low accuracy. The second dataset was collected using the longer range setting for comparison.

5.4 Evaluation Method and Ground Truth

Obtaining ground truth for underground datasets was difficult due to large environments with breaks in line-of-sight and the absence of GPS signal. In order to properly evaluate the accuracy of the generated maps and the relative localization, the surveyed positions of reference markers in the subway system are used as ground truth. The distance between

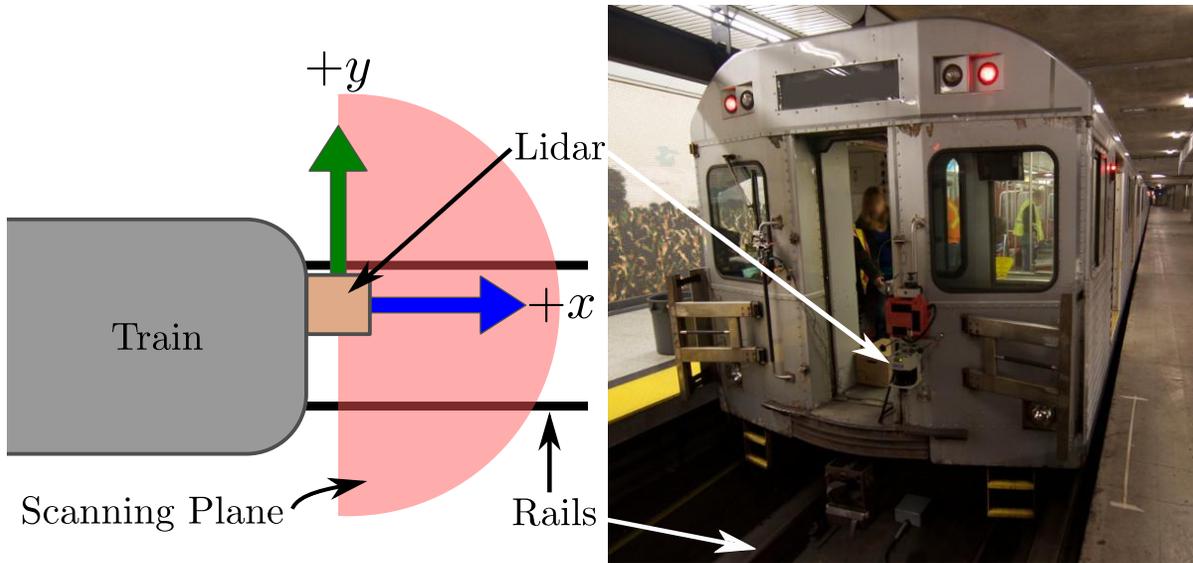


Figure 5.4: *Right*: Picture of the train used for the experimental work. The 2D lidar is seen in the middle of the picture attached to the front handrail of the train. *Left*: Overhead view of the sensor installation with respect to the train.

reference markers was obtained from signalling drawings, which specified the location of the reference markers based on physical surveys that have been performed in the subway tunnels. The reference markers are the locations of signal posts used to signal whether it is safe to enter the next section of track or if it is occupied by another train. They define the boundaries of the *fixed blocks* in the signalling system. A picture of a reference marker is labelled in the image of a tunnel environment in Figure 5.1 and they are also labelled in the generated maps in Figure 5.2.

A reference marker location is manually assigned to a scan in which it appears, each reference marker is assigned to one scan per run. The manual detection was completed by simultaneously viewing the stream of lidar scans and camera images from the USB webcam that was also mounted on the sensor head. The images helped identify when a signal post was approaching because the signals are red or green lights that are clearly visible. The location within the scan where the signal post was located was also recorded to minimize the error due to the manual marking of ground truth. The global accuracy of the reference markers is subject to many errors (manual identification, surveying, etc.)

Table 5.1: October 26th, 2014 Dataset Information

Run #	Travel Direction	# of Reference Markers	Max Speed [km/h]	Duration [s]
1	Backward	31 ^a	70	509
2	Forward	33	70	610
3	Backward	32	70	564
4	Forward	33	15	1864
5	Backward	32	35	746
6	Forward	33	35	943

^aDuring Run 1, the first reference marker was not captured due to data logging issues.

Table 5.2: November 30th, 2014 Dataset Information

Run #	Travel Direction	# of Reference Markers	Max Speed [km/h]	Duration [s]
7	Backward	31	70	525
8	Forward	32	70	652
9	Backward	29 ^a	35	742
10	Forward	32	35	917
11	N/A ^b			
12	Backward	31	35	722
13	Forward	32	70	637
14	Backward	31	70	520
15	Forward	32	70	606

^aDuring Run 9, the last two reference markers were not captured due to sensor issues.

^bDuring Run 11, the train did not leave the station due to other tests, this run is not analyzed.

and could be on the order of metres but only affects the mapping accuracy. The indexed scans were compared manually to ensure the identical physical locations were selected for a specific reference marker in all runs in order to minimize relative error between runs. Thus, the relative accuracy of the reference markers is within centimetres for relative localization.

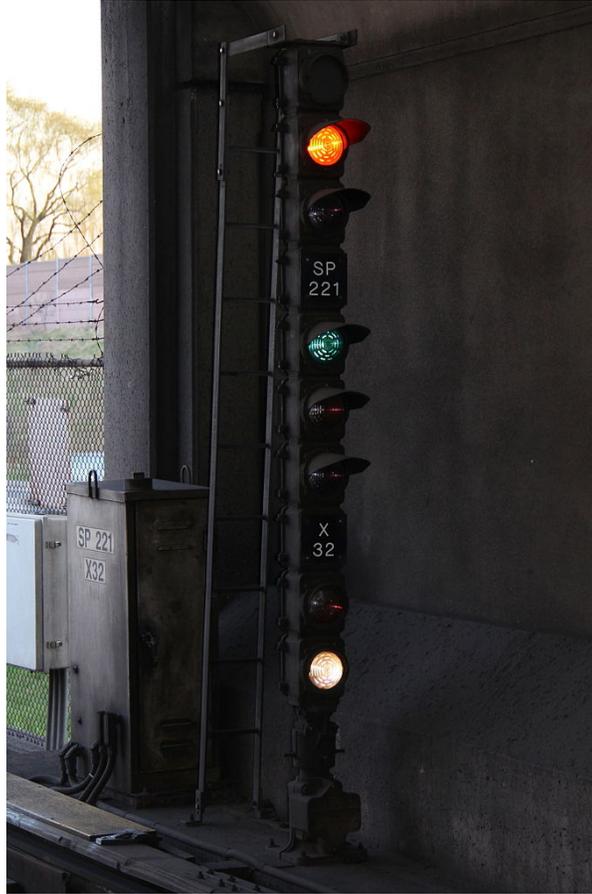


Figure 5.5: Image of a signal post used as a reference marker. The location of these reference markers were provided and could be used as ground truth for localization and mapping. Image credit: Kevin Hadley

5.5 Summary

In this Chapter, the experimental setup used to collect datasets for algorithm development and testing is described. A total of 14 datasets were collected along two parallel tracks of underground tunnel that are 5.3km long with varying maximum train speed, ranging from 15 to 70 km/h. This allows for the creation of 14 maps and 98 unique combinations of possible runs for localization. The tunnel environments and experimental setup used to gather the datasets is also explained. This section also describes the method used to obtain ground truth data for evaluating the results obtained in the following chapter.

Chapter 6

Experimental Results

The performance of the system was evaluated for both mapping and localization accuracy. The two phases are evaluated separately due to the nature of their errors. The mapping phase uses dead-reckoning to generate a locally consistent map, therefore the errors are reported as cumulative error as well as percentage based on the segment lengths between reference markers. The localization phase provides position estimates relative to the map, thus the relative error at the reference markers are reported.

6.1 Mapping

Trajectory estimates were calculated for each of the runs listed in Tables 5.1 and 5.2. Figure 6.3 shows the result of the odometry estimation used to create the map for Run 7 and Figure 6.4 shows the same for Run 3. The estimated train trajectory is plotted along with the reference markers as well as the errors at each reference marker. This error data were collected for each run and used to create the plots in Figures 6.2 and 6.1.

Error percentages are calculated based on the segment lengths between reference markers and the error calculated at the reference marker at the end of the segment. This allows for the comparison of mapping error based on the tunnel environment of the segment to highlight potential strengths and weaknesses. Subdividing each run also

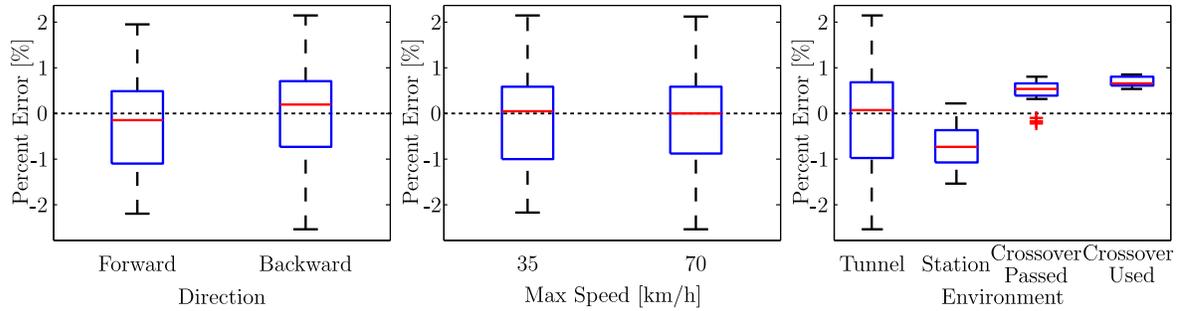


Figure 6.1: Mapping results for Runs 7 to 15 separated by train directions, maximum speeds and tunnel environments. Errors are reported with respect to the reference markers.

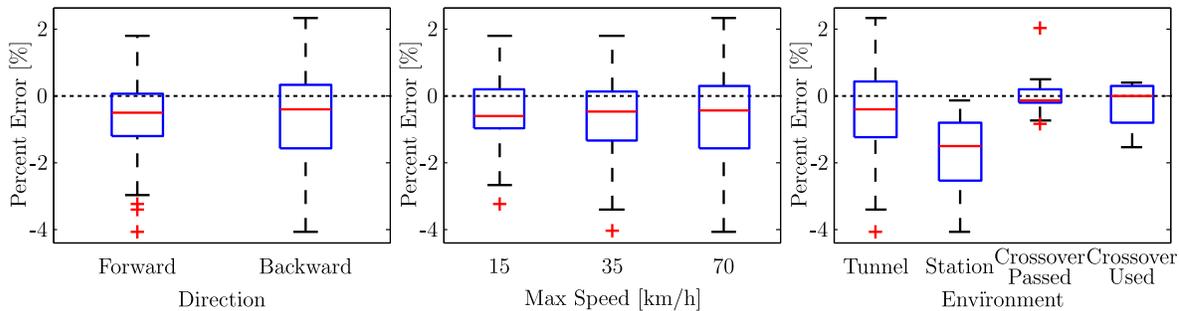


Figure 6.2: Mapping results for Runs 1 to 6 separated by train directions, maximum speeds and tunnel environments. Errors are reported with respect to the reference markers.

provides many data points per run increasing the significance of the resulting trends and it also removes some of the averaging out effect of random noise. The error metric used for comparison will be the percent error:

$$\%e = \frac{e}{l}, \quad (6.1)$$

where e is the error for a segment and l is the segment length. The overall performance of the mapping phase can be seen in the left box plots. The average error is around -0.6% in both directions, which means the maps are a bit short. This bias towards a short map is due to small segments of surfaces, which are near parallel to the direction of travel, that are below the threshold for removal during pre-processing. These segments,

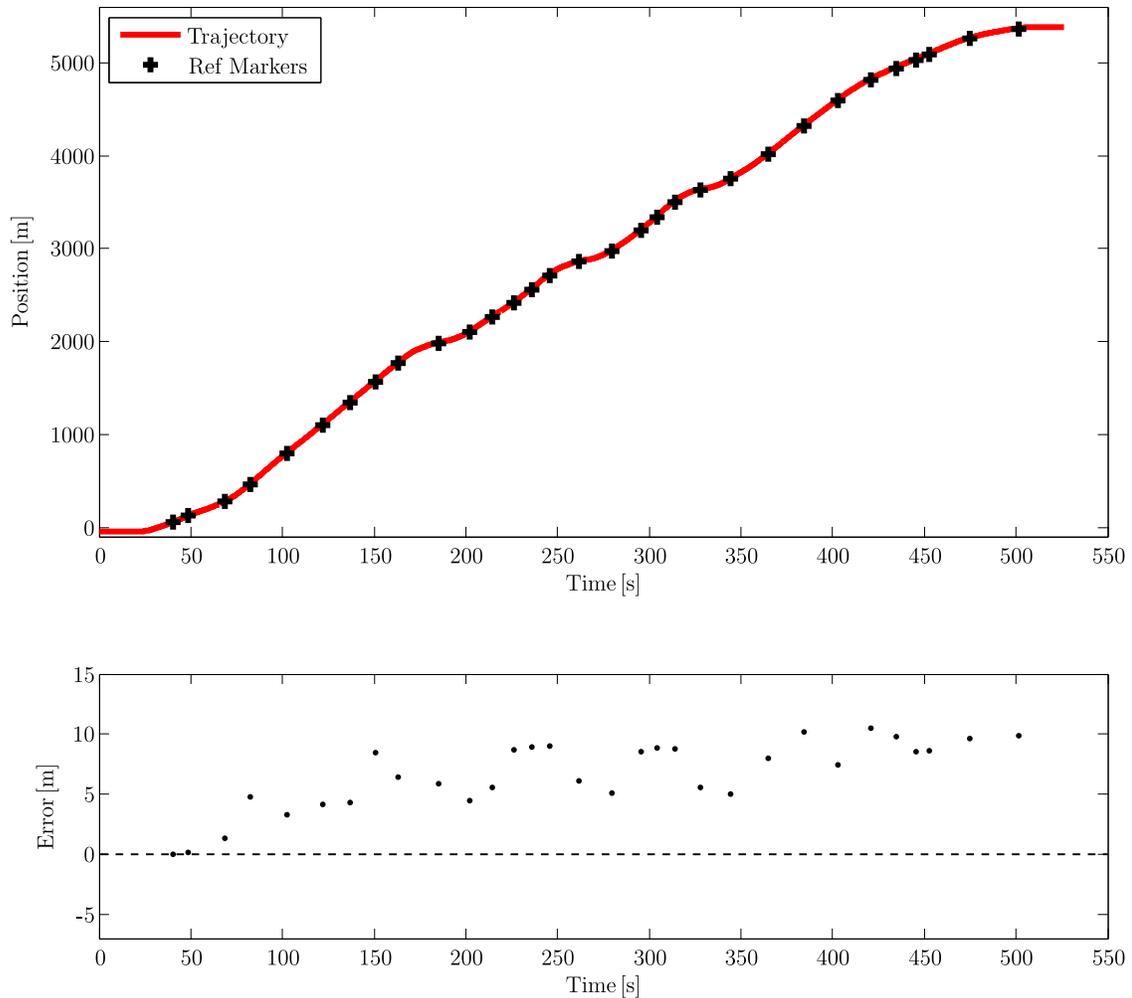


Figure 6.3: Mapping result presenting estimated trajectory. Top: Estimated trajectory of the train during the mapping phase for Run 7. The reference markers are shown at their surveyed locations as the time the train passed them. Bottom: Estimated error at each reference marker. This shows the accuracy of the mapping phase with respect to the location of the known reference markers in the tunnel. It is clear that the error grows unbounded with distance, but at the low rate of approximately 0.2% by distance.

especially when close to the lidar (more data points and closer together), cause a bias for underestimating the distance travelled between scans. The middle box plots show the similarity in accuracy over a wide range of speeds. The apparent increase in variance with speed is caused by the increase in uneven number of runs for each maximum speed. The right box plots in the figures show the station environment has the largest error, which is caused by two issues. First, the train stopped or slowed significantly at every station

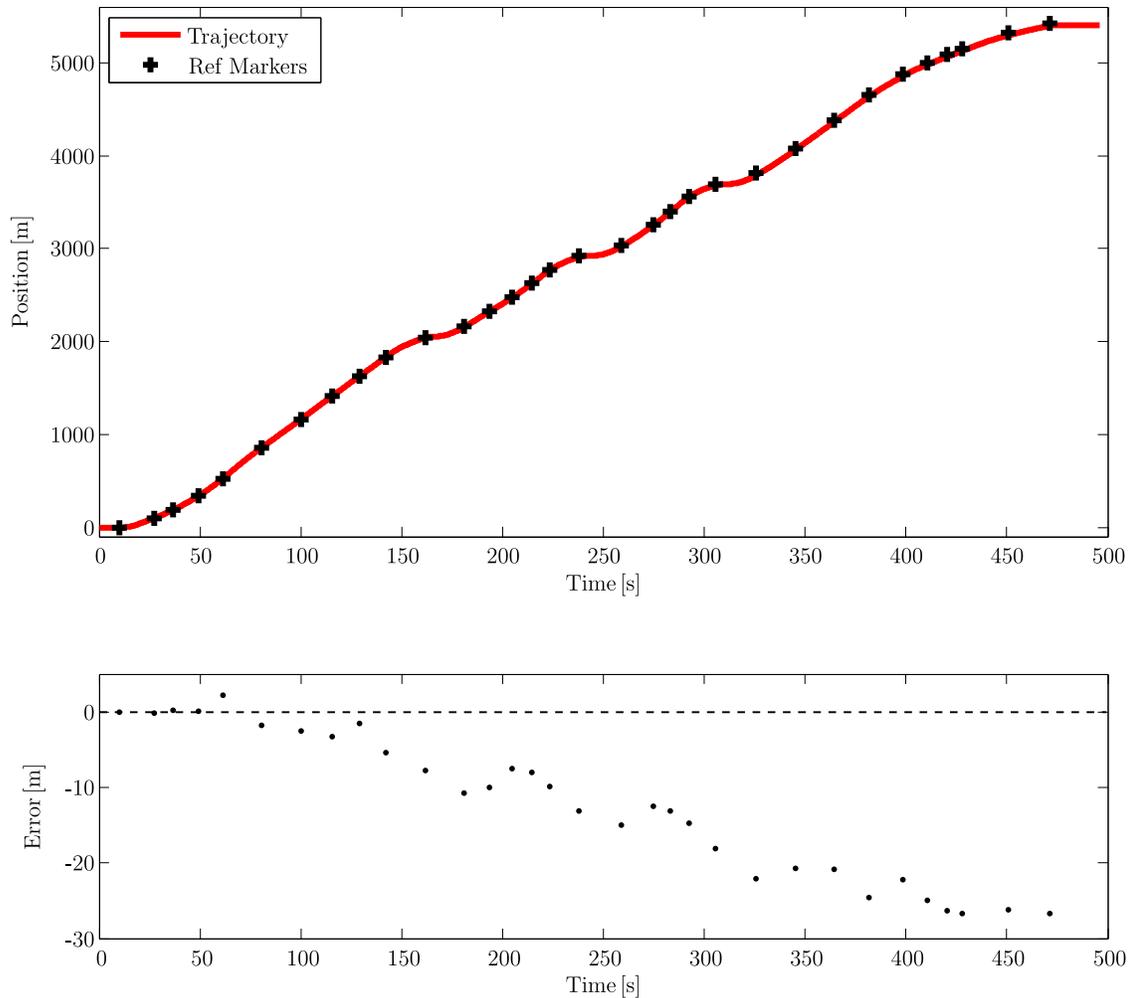


Figure 6.4: Mapping result presenting estimated trajectory. Top: Estimated trajectory of the train during the mapping phase for Run 3. The reference markers are shown at their surveyed locations as the time the train passed them. Bottom: Estimated error at each reference marker. This shows the accuracy of the mapping phase with respect to the location of the known reference markers in the tunnel. It is clear that the error grows unbounded with distance, but at the low rate of approximately -0.6% by distance.

resulting in larger accelerations than at other portions of the railway. This contrasts the constant-acceleration assumptions used in the odometry algorithm. Second, this environment has many small near-parallel surfaces that lead to a negative odometry bias as explained previously.

Figure 5.2 demonstrates the quality of the generated maps and confirms the accuracy of the odometry used to generate the maps. The entire map for Run 3 is shown for scale

and subsections of the map are enlarged to highlight the crispness of the map in each tunnel environment.

6.2 Localization

In order to maximize the utility of the datasets collected, maps were constructed from each run and then each run was localized against all possible maps. This made it possible to test the system over 98 unique combinations (49 forwards and 49 backwards) for a total of over 500 km. Tables 6.1 and 6.2 summarize the results obtained from these tests localizing short range data to short range maps and likewise for the long range data. The maximum and median error values, in centimetres, are calculated at the ground truth locations on the track. These errors are based on the delta in the estimated locations of the reference markers between the mapping and localizations phases. The estimated locations of the reference markers are based on positions assigned to the scans identified as containing them. In both the mapping and localization phases, scan positions are recorded relative to the first scan in the map dataset.

Table 6.1: Localization results for the short range sensor datasets.

Backward Runs - Errors in centimetres						
Run #	Map 1		Map 3		Map 5	
	max	med	max	med	max	med
1	4.6	1.4	5.6	2.0	6.1	1.6
3	6.8	1.4	4.2	1.3	29.0	1.4
5	10.1	2.3	29.0	1.8	3.8	1.8

Forward Runs - Errors in centimetres						
Run #	Map 2		Map 4		Map 6	
	max	med	max	med	max	med
2	8.2	1.6	32.2	3.5	48.6	2.5
4	17.7	3.7	7.3	2.3	19.1	2.9
6	48.2	2.9	16.3	2.7	7.5	1.8

Table 6.2: Localization results for the long range sensor datasets.

Backward Runs - Errors in centimetres									
Run #	Map 7		Map 9		Map 12		Map 14		
	max	med	max	med	max	med	max	med	
7	8.8	1.8	11.0	5.0	11.6	2.1	11.8	3.9	
9	15.0	5.6	8.1	2.5	14.5	3.6	12.9	3.6	
12	17.7	4.3	13.9	5.0	7.5	1.7	14.7	3.1	
14	15.2	4.5	11.0	2.8	15.3	3.8	8.0	1.2	

Forward Runs - Errors in centimetres									
Run #	Map 8		Map 10		Map 13		Map 15		
	max	med	max	med	max	med	max	med	
8	5.7	1.5	81.8	3.9	12.5	4.8	19.9	6.1	
10	15.1	4.2	82.9	1.1	10.6	3.1	77.9	4.1	
13	15.5	3.8	84.0	4.5	6.6	1.6	16.9	4.2	
15	25.2	4.8	16.7	5.3	16.9	5.0	74.4	1.5	

The maximum and median errors are presented to show the error spread for each run. As expected, localizing using the same data that was used to create the map results in very small errors, which can be seen on the diagonals of Tables 6.1 and 6.2. Figures 6.5 and 6.6 are a histogram of the errors used to create these tables (note the logarithmic scale on the y -axis). The histograms show that most of the errors are centred near zero with the exception of a few outliers. The outliers near ± 70 cm are caused by an

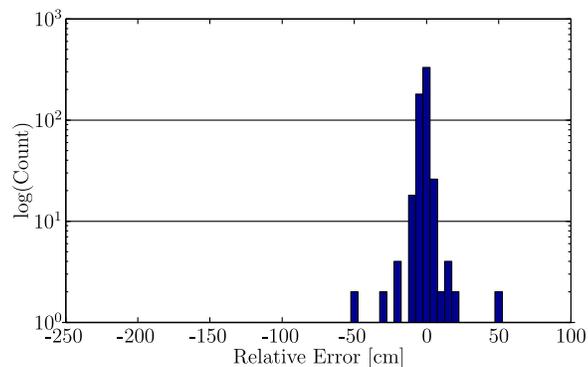


Figure 6.5: Localization results. Histogram of the relative errors used in Table 6.1. Note the logarithmic scale on the y -axis.

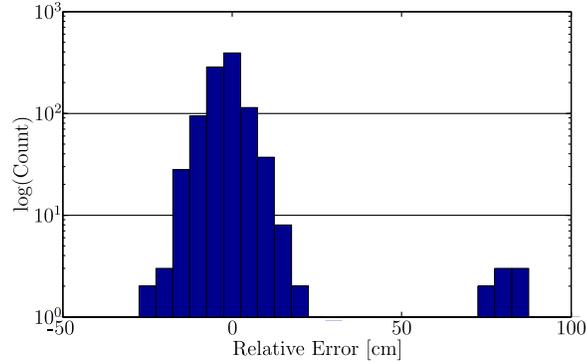


Figure 6.6: Localization results. Histogram of the relative errors used in Table 6.2. Note the logarithmic scale on the y -axis.

association error of the handle rails in the tunnel, as the left and right hand rails of the escape ladders are about 70 cm apart.

Table 6.3: Long range sensor localization results using the short range datasets for mapping.

Backward Runs - Errors in centimetres						
Run #	Map 1		Map 3		Map 5	
	max	med	max	med	max	med
7	43.6	5.4	33.8	4.5	33.2	4.9
9	46.3	4.9	41.8	4.6	33.1	4.1
12	41.4	3.7	45.4	3.6	80.0	4.0
14	46.0	5.0	36.4	4.8	32.2	4.8

Forward Runs - Errors in centimetres						
Run #	Map 2		Map 4		Map 6	
	max	med	max	med	max	med
8	37.7	6.0	39.3	7.9	40.2	6.6
10	34.4	5.4	47.8	8.5	42.8	5.1
13	29.2	5.6	40.3	8.9	30.6	7.8
15	19.9	4.1	43.6	6.4	39.6	4.8

Tables 6.3 and 6.4 summarize the results of localization using one sensor mode to generate a map and the other to localize against those maps. The results of mixing the two maximum range modes of the lidar sensor can be seen to increase both the maximum and median errors for localization. The worst performance occurs when the long range

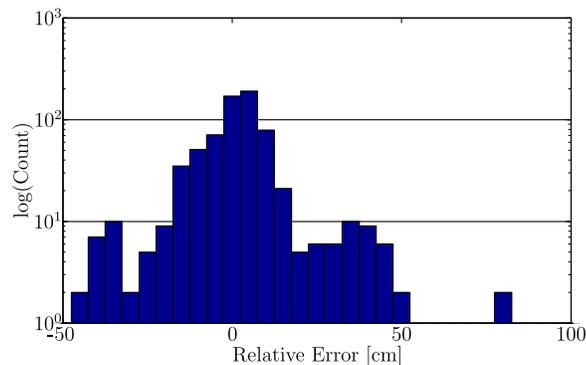
Table 6.4: Short range sensor localization results using the long range datasets for mapping.

Backward Runs - Errors in centimetres									
Run #	Map 7		Map 9		Map 12		Map 14		
	max	med	max	med	max	med	max	med	
1	44.3	7.6	195.4	8.3	42.5	6.0	47.6	6.8	
3	230.4	9.6	46.2	7.1	44.1	6.4	46.2	7.7	
5	48.9	8.7	47.9	7.1	220.0	7.2	52.8	8.0	

Forward Runs - Errors in centimetres									
Run #	Map 8		Map 10		Map 13		Map 15		
	max	med	max	med	max	med	max	med	
2	37.5	6.1	43.4	6.5	35.5	7.6	38.9	5.9	
4	42.8	9.6	43.9	11.3	43.8	10.7	44.9	8.7	
6	44.1	9.3	35.9	10.9	33.0	9.1	33.9	8.2	

setting data are used to generate the map and the short range setting data are used for localization. The long range setting data increases map density and possibly includes of parts of the tunnel which are not observed when using the short range setting. This is a possible cause of the increased error in localization, because the short ranged data used for localization contains less data compared to the more data rich map.

This histograms in Figures 6.7 and 6.8 show the results used to create Tables 6.3 and 6.4 respectively. These figures show that the error is again centred about zero.

Figure 6.7: Localization results. Histogram of the relative errors used in Table 6.3. Note the logarithmic scale on the y -axis.

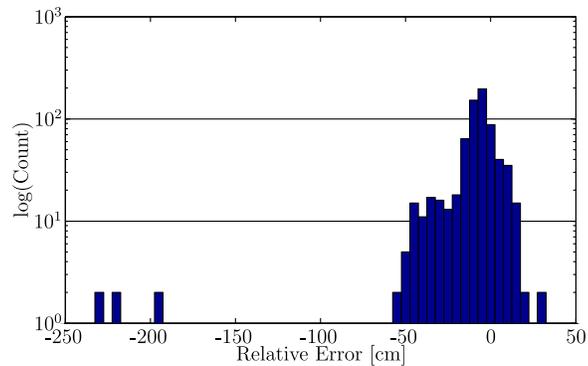


Figure 6.8: Localization results. Histogram of the relative errors used in Table 6.4. Note the logarithmic scale on the y -axis.

There now appears to be more of a bimodal distribution in the error, but the extent of this partially a visual effect of the logarithmic scale on the y -axis. The very low frequency of outliers is also highlighted by these figures.

The relatively large maximum errors seen in the above tables are the result of convergence to the incorrect local minimum of the RQE cost function, which happens from time to time. This is often the result of the pattern of handrails seen in the tunnel map in Figure 5.2 and the small differences in odometry between the mapping and localization phases that lead to initial position estimates near a local minima. This is demonstrated by Figure 6.9, which is a histogram of all the errors used to create Tables 6.1 to 6.4. Here the majority of the large errors are located around ± 70 cm, which is the distance between a set of handrails. The histogram also shows that over 90% of the errors are less than 5 cm.

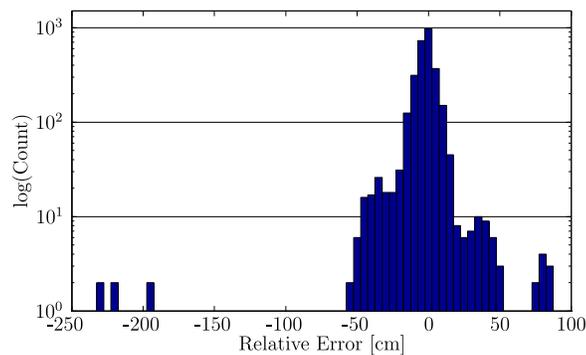


Figure 6.9: Localization results. Histogram of the relative errors used in from all localization runs. Note the logarithmic scale on the y -axis.

6.3 Summary

In this chapter, the performance of the proposed mapping and localization system are evaluated using the datasets and ground truth collected, as described in the previous chapter. The mapping and localization algorithms are evaluated separately because the first generates a globally consistent map of the environment, while the second provides localization relative to the previously generated map. The results demonstrate that the mapping accuracy is between 0.2 and 0.6% error by distance and the localization accuracy is within ± 10 cm in stations in over 90% of the tunnel with a relatively normal distribution over a range of ± 10 cm and outliers of up to 80 cm in most cases and 2.3 m in the case when the short range sensor setting was used for localization and the long range for generating the map.

Chapter 7

Discussion and Future Work

After the first datasets were collected it became apparent that wall roughness in the tunnels was not reliable for localization as initially predicted. Thus, it was concluded that using the short range setting of the SICK LMS 291 for higher accuracy was not ideal in the tunnel environment. Instead, other objects and infrastructure in the tunnel needed to be used for localization. However, when the long-range setting of the lidar sensor was used to detect non parallel wall aspects of the tunnel, as far in front of the sensor as possible to minimize the drop outs, the localization error did not significantly decrease.

Although the train in the experiments does not technically qualify as a ‘high-speed’ train, which is defined as a train capable of travelling over roughly 200 km/h, from a robotics perspective, this is a high-speed application. Initially, there was concern about the effects of motion distortion on the lidar scans, as each data point is registered to a scan at different times while the sensor is moving. However, it was found that motion distortion did not significantly impact the system for two reasons. First, motion distortion does not affect the identification of near-parallel walls in the pre-processing stage. This is due to the fact that the perceived angle of these walls is not affected by motion distortion because they are nearly parallel to the direction of travel. Second, the odometry algorithm

uses consecutive scans gathered over a short period of time that are affected by similar amounts of motion distortion due to the limited acceleration of the train. Initial testing shows that the system performs well up to 70 km/h. Modern trains can travel over 400 km/h, speeds that are generally associated with severe motion distortion for sensors like a 2D lidar. However, at those high speeds motion compensation would not likely be required. The remaining data after pre-processing only occupies a small field of the sensor's scan range, generally only a few degrees of the field of view, meaning the impacts of motion distortion are limited. As a result, the system is considered robust to the effects of motion distortion.

Prior to implementing the system, it will need to be tested for robustness against noise and dynamic environments. Although underground tunnels rarely change, station platforms and any outdoor environments in a railway system will contribute to sensor noise and create a dynamic environment. Sensor configuration and location-specific algorithm settings can be employed to avoid or ignore some sources of noise and environment change. However, weather, dust, and construction cannot be avoided and may affect localization. Small amounts of sensor noise should not affect the system as the sliding-window filter uses many scans that presumably would not be affected by the same sensor noise. Also, the RQE-based scan matching was not significantly affected by isolated data points or groups of points that occur in isolation only in one scan. The sliding-window portion of the system would only be affected by rapid dynamic changes, as the sliding windows are temporally short. The optimization of the sliding window for a constant acceleration over the window is a simplified version of continuous-time trajectory estimation Furgale et al. (2015); Anderson et al. (2015). Further improvements may be realized by exploring a more complex continuous-time trajectory estimation implementation.

This system operates on prior knowledge about starting conditions such as location, speed, and in the case of localization, the railway to be traversed. The initial velocity is easily recovered from the odometry algorithm. However, the system cannot solve the

kidnapped-robot problem (i.e., determine which railway it is on and where along the railway it is located without any prior knowledge). Solving this problem would involve attempting to match the current scan to all locations in approximately 0.5 m increments across the entire network of maps, which is computationally prohibitive in the system's current form.

The proposed method currently finds and removes parallel walls but does nothing else with this information. Information about the walls could be used to increase localization accuracy or speed. For example, the presence or absence of walls on each side, their distance from the sensor and the width of the tunnel would provide a low dimensional source of data that could be easily incorporated into the localization phase, or to solve the kidnapped-robot problem. The RQE point-cloud alignment tuning parameter σ , from Equation 3.6, is currently determined empirically and remains fixed. However, σ should be based on the uncertainty of the location of a data point as well as the underlying surface from which it was sampled. Thus, σ could be made a function of several parameters; however, ideally it would be weighted by the data point's range from the sensor because the distance between sampled points increases as range increases and generally the noise increases as well.

The algorithms presented in this thesis were implemented in Matlab and run several times slower than real time. It is possible to optimize the RQE-based objective function using Improved Fast Gauss Transforms (IFGT) to reduce the problem from a quadratic computation cost to a linear computation cost, as presented by Sheehan et al. (2013). Another solution would be to down-sample the window and local submap point clouds to reduce the data overlap in the point clouds prior to matching them. Developing a linear point density filter is a logical next step towards implementing this algorithm, as the generated maps consist almost exclusively of straight lines which contain many overlapping data points. With a properly implemented density filter, additional scans could be used to generate the map, which would avoid missing small details in the map.

The mapping algorithm currently uses a subsets of scans that are spaced every 0.5 m, based on lidar odometry, and disregards all scans in between. This method was selected because it was empirically found to be an acceptable balance between map density and data loss. However, a linear density filter could provide significant improvements both in terms of eliminating redundant data points in the map and including information currently lost by the scan down sampling. Finally, the use of a graphics processor unit (GPU) for RQE function evaluations should be investigated, as it may significantly reduce computation times.

Chapter 8

Conclusion

In this thesis, a framework for estimating the position and velocity of a rail-based vehicle in a tunnel is presented. The framework consists of two phases, a mapping phase and a localization phase. The presented algorithms were tested on 75 km of data gathered across 14 runs in a 5.3 km section of underground subway tunnel. By combining an RQE-based point-cloud alignment algorithm with a sliding-window filter for odometry, it has been shown that it is possible to robustly achieve low relative localization and absolute mapping errors. The subway tunnel can be mapped with less than 0.6% error over the total length of the generated maps. The algorithm is capable of continuously localizing, relative to the generated map, to within 10 cm in stations and at crossovers, and 2.3 m in pathological sections of tunnel. The algorithms developed and the results they generated using the first six “short range” datasets have been peer reviewed and published in (Daoust et al., 2016).

The challenging tunnel environment and high-speed nature of underground railway travel led to two key findings. First, using RANSAC to find and remove tunnel walls that are near-parallel to the direction of travel of the train highlights the data that changes from scan to scan and removes the data that is not informative. This point was necessary in order for the algorithm to function correctly. Second, the odometry algorithm robustly

deals with the noise and data gap issues that arise when a significant portion of data from lidar scan has been removed.

The identification and removal of planar surfaces that are near parallel to the motion of travel from geometry-based localization data is an important take-away from this thesis. This concept can be used to help downsample large datasets and highlight data that provides greater accuracy for localization algorithms while decreasing computational cost of processing the datasets.

In summary, the novel contributions of this thesis are as follow. First, the development of an infrastructure-free mapping algorithm for use in tunnel environments at high speed. Second, the development of an infrastructure-free localization algorithm for high-speed rail vehicles in tunnels. Third, Experimental validation of both algorithms on a subway train in a real tunnel with over 98 km of data.

Bibliography

- Acharya, A., Sadhu, S., and Ghoshal, T. (2011). Train localization and parting detection using data fusion. *Transportation Research Part C: Emerging Technologies*, 19(1):75–84.
- Albrecht, T., Luddecke, K., and Zimmermann, J. (2013). A precise and reliable train positioning system and its use for automation of train operation. In *Intelligent Rail Transportation (ICIRT), 2013 IEEE International Conference on*, pages 134–139. IEEE.
- Anderson, S., Barfoot, T. D., Tong, C. H., and Särkkä, S. (2015). Batch nonlinear continuous-time trajectory estimation as exactly sparse gaussian process regression. *Autonomous Robots*.
- Besl, P. and McKay, H. (1992). A method for registration of 3-D shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256.
- Beugin, J. and Marais, J. (2012). Simulation-based evaluation of dependability and safety properties of satellite technologies for railway localization. *Transportation Research Part C: Emerging Technologies*, 22:42–57.
- Boehringer, F. (2003). Train location based on fusion satellite and train-borne sensor data. In *AeroSense 2003*, pages 76–85. International Society for Optics and Photonics.
- Boehringer, F. and Geistler, A. (2006). Comparison between different fusion approaches

- for train-borne location systems. In *Multisensor Fusion and Integration for Intelligent Systems, 2006 IEEE International Conference on*, pages 267–272. IEEE.
- Bonnett, C. F. (2005). *Practical railway engineering*. World Scientific.
- Bosse, M. C. (2004). *ATLAS: a framework for large scale automated mapping and localization*. PhD thesis, Massachusetts Institute of Technology.
- Chen, Y. and Medioni, G. (1991). Object modeling by registration of multiple range images. In *Robotics and Automation, 1991. Proceedings of the IEEE International Conference on*, pages 2724–2729. IEEE Comput. Soc. Press.
- Corsino Espino, J., Stanciulescu, B., and Forin, P. (2013). Rail and turnout detection using gradient information and template matching. In *Intelligent Rail Transportation (ICIRT), 2013 IEEE International Conference on*, pages 233–238. IEEE.
- Daoust, T., Pomerleau, F., and Barfoot, T. D. (2016). Light at the end of the tunnel: High-speed, lidar-based train localization in challenging underground environments. In *Computer and Robot Vision (CRV), Canadian Conference on, IEEE*.
- Elseberg, J., Borrmann, D., and Nüchter, A. (2013). Automatic and Full Calibration of Mobile Laser Scanning Systems. In *Experimental Robotics*, pages 907–917. Springer.
- Ernest, P., Mazl, R., and Preucil, L. (2004). Train locator using inertial sensors and odometer. In *IEEE intelligent vehicles symposium*, pages 860–865.
- Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- Furgale, P., Tong, C. H., Barfoot, T. D., and Sibley, G. (2015). Continuous-time batch trajectory estimation using temporal basis functions. *IJRR*.

- Geistler, A. and Bohringer, F. (2004). Robust velocity measurement for railway applications by fusing eddy current sensor signals. In *Intelligent Vehicles Symposium, 2004 IEEE*, pages 664–669. IEEE.
- Goodin, C., Durst, P. J., Prevost, Z. T., and Compton, P. J. (2013). A probabilistic model for simulating the effect of airborne dust on ground-based LIDAR. In *SPIE Defense, Security, and Sensing*, pages 87340D–87340D. International Society for Optics and Photonics.
- Hahnel, D., Burgard, W., Fox, D., and Thrun, S. (2003). An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 1, pages 206–211. IEEE.
- Hasberg, C., Hensel, S., and Stiller, C. (2012). Simultaneous localization and mapping for path-constrained motion. *Intelligent Transportation Systems, IEEE Transactions on*, 13(2):541–552.
- Heirich, O., Robertson, P., Garcia, A. C., and Strang, T. (2012a). Bayesian train localization method extended by 3D geometric railway track observations from inertial sensors. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 416–423. IEEE.
- Heirich, O., Robertson, P., García, A. C., Strang, T., and Lehner, A. (2012b). Probabilistic localization method for trains. In *Intelligent Vehicles Symposium (IV), 2012 IEEE*, pages 482–487. IEEE.
- Heirich, O., Robertson, P., and Strang, T. (2013a). RailSLAM-Localization of rail vehicles and mapping of geometric railway tracks. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 5212–5219. IEEE.

- Heirich, O., Steingass, A., Lehner, A., and Strang, T. (2013b). Velocity and location information from onboard vibration measurements of rail vehicles. In *Information Fusion (FUSION), 2013 16th International Conference on*, pages 1835–1840. IEEE.
- Hensel, S. and Hasberg, C. (2009). Bayesian techniques for onboard train localization. In *Statistical Signal Processing, 2009. SSP'09. IEEE/SP 15th Workshop on*, pages 361–364. IEEE.
- Hensel, S. and Hasberg, C. (2010). Probabilistic landmark based localization of rail vehicles in topological maps. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 4824–4829. IEEE.
- Hensel, S., Hasberg, C., and Stiller, C. (2011). Probabilistic rail vehicle localization with Eddy current sensors in topological maps. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1525–1536.
- Javed, A., Zhang, H., Huang, Z., and Deng, J. (2013). BWS: Beacon-driven wake-up scheme for train localization using wireless sensor networks. Technical report, Technical Report OUCS-2013–13, University of Otago.
- Jose, E., Adams, M., Mullane, J. S., and Patrikalakis, N. M. (2010). Predicting Millimeter Wave Radar Spectra for Autonomous Navigation. *Sensors Journal, IEEE*, 10(5):960–971.
- Kiefer, J. (1953). Sequential minimax search for a maximum. *Proceedings of the American mathematical society*, 4(3):502–506.
- Lauer, M. and Stein, D. (2013). Algorithms and concepts for an onboard train localization system for safety-relevant services. In *Intelligent Rail Transportation (ICIRT), 2013 IEEE International Conference on*, pages 65–70. IEEE.

- Lauer, M. and Stein, D. (2015). A train localization algorithm for train protection systems of the future. *Intelligent Transportation Systems, IEEE Transactions on*, 16(2):970–979.
- Lingemann, K., Surmann, H., Nuchter, A., and Hertzberg, J. (2004). Indoor and outdoor localization for fast mobile robots. In *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, volume 3, pages 2185–2190. IEEE.
- Maddern, W., Harrison, A., and Newman, P. (2012). Lost in translation (and rotation): Rapid extrinsic calibration for 2D and 3D LIDARs. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 3096–3102. IEEE.
- Manz, H., Rtters, R., Schfer, B., and Engelhardt, T. (2011). Metrological standardization for rail applications of satellite based localization systems.
- Marshall, J., Barfoot, T., and Larsson, J. (2008). Autonomous underground tramming for center-articulated vehicles. *Journal of Field Robotics*, 25(6-7):400–421.
- Mazl, R. and Přeučil, L. (2003). Sensor data fusion for inertial navigation of trains in gps-dark areas. In *Intelligent Vehicles Symposium, 2003. Proceedings. IEEE*, pages 345–350. IEEE.
- Merali, R. S., Tong, C. H., Gammell, J., Bakambu, J., Dupuis, E., and Barfoot, T. D. (2012). Three-Dimensional Surface Mapping Using a Semi-Autonomous Rover: A Planetary Analogue Field Experiment. In *Proceedings of the 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space (iSAIRAS)*, Turin, Italy.
- Milford, M. (2013). Vision-based place recognition: how low can you go? *The International Journal of Robotics Research*, 32(7):766–789.

- Milford, M. J. and Wyeth, G. F. (2012). Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE.
- Nüchter, A., Elseberg, J., and Borrmann, D. (2013). Irma3D - an Intelligent Robot for Mapping Applications. In *Proceedings of the 3rd IFAC Symposium on Telematics Applications (TA '13)*, pages 119–124. IFAC.
- Nüchter, A., Elseberg, J., Schneider, P., and Paulus, D. (2010). Study of parameterizations for the rigid body transformations of the scan registration problem. *Computer Vision and Image Understanding*, 114(8):963–980.
- Nüchter, A., Lingemann, K., and Hertzberg, J. (2006). 6D SLAM with Kurt3D. *Robotic 3D Environment Cognition*, page 59.
- Nüchter, A., Lingemann, K., Hertzberg, J., and Surmann, H. (2007). 6D SLAM3D mapping outdoor environments. *Journal of Field Robotics*, 24(8-9):699–722.
- Shafia, M. A., Sadjadi, S. J., Jamili, A., Tavakkoli-Moghaddam, R., and Pourseyed-Aghaee, M. (2012). The periodicity and robustness in a single-track train scheduling problem. *Applied Soft Computing*, 12(1):440–452.
- Sheehan, M., Harrison, A., and Newman, P. (2011). Self-calibration for a 3D laser. *The International Journal of Robotics Research*, page 0278364911429475.
- Sheehan, M., Harrison, A., and Newman, P. (2013). Continuous vehicle localisation using sparse 3D sensing, kernelised Rényi distance and fast Gauss transforms. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pages 398–405. IEEE.
- Shenton, R. (IRSE ASPECT, London, 26 September 2008). Train video positioning.
- Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic robotics*. MIT press.

- Thrun, S., Fox, D., Burgard, W., and Dellaert, F. (2001). Robust monte carlo localization for mobile robots. *Artificial intelligence*, 128.
- Tong, C. H., Anderson, S., Dong, H., and D Barfoot, T. (2014). Pose interpolation for laser-based visual odometry. *Journal of Field Robotics*, 31(5):731–757.
- Tong, C. H., Barfoot, T. D., and Dupuis, E. (2011). 3D SLAM for planetary worksite mapping. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 631–638. IEEE.
- Widmann, G. R., Daniels, M. K., Hamilton, L., Humm, L., Riley, B., Schiffmann, J. K., Schnelker, D. E., and Wishon, W. H. (2000). *Comparison of lidar-based and radar-based adaptive cruise control systems*. SAE International.
- Wohlfeil, J. (2011). Vision based rail track and switch recognition for self-localization of trains in a rail network. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 1025–1030. IEEE.
- Zhang, J. and Singh, S. (2014). Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems Conference (RSS)*.
- Zlot, R. and Bosse, M. (2009). Place recognition using keypoint similarities in 2D lidar maps. In *Experimental Robotics*, pages 363–372. Springer.
- Zlot, R. and Bosse, M. (2014). Efficient large-scale 3d mobile mapping and surface reconstruction of an underground mine. In *Field and service robotics*, pages 479–493. Springer.
- Zuliani, M. (2008). Ransac toolbox for matlab. *MATLAB Central*.