Master's Thesis

Underground Representations for Robot Localization and Mapping

Alexander Baikovitz

CMU-RI-TR-21-42

June 2021

Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Michael Kaess, Chair
Dimitrios Apostolopoulos
Paloma Sodhi

Submitted in partial fulfillment of the requirements for the degree of Master of Science.

Keywords: SLAM; localization; ground penetrating radar

Abstract

There has been exciting recent progress in using radar as a sensor for robot navigation given its increased robustness to varying environmental conditions. However, within these different radar perception systems, ground penetrating radar (GPR) remains under-explored. By measuring structures beneath the ground, GPR can provide stable features that are less variant to ambient weather, scene, and lighting changes, making it a compelling choice for long-term spatio-temporal mapping.

In this work, we present a set of approaches for robots to naturally reason about subsurface information for robust localization and mapping in unknown environments. First, we propose a novel method for place recognition using GPR measurements for robot localization. We achieve this by horizontally stacking one-dimensional GPR measurements into two-dimensional images. This is followed by a spatial correlation network over learned image features to correct for translational drift. We find that this approach improves GPR-based localization performance compared to engineered heuristics as it can learn distinct features in often noisy and repetitive images.

Second, we propose a GPR-based simultaneous localization and mapping (SLAM) method that does not require re-visitation by explicitly modeling the relationships among linear features, such as pipes and geologic fractures, commonly found in real-world scenes. We formulate this as inference over a factor graph to jointly estimate latent Hough line features and robot states. We find that this approach effectively reduces drift perpendicular to line observations and allows us to simultaneously reconstruct the underground environment features and estimate the robot state.

Acknowledgments

I first would like to thank Michael Kaess for his continued support, detailed feedback, and late night paper formatting edits. Thank you Dimi Apostolopolous for supporting my research journey and providing me with my first GPR system, which enabled me to explore this exciting sensing modality. Thank you Paloma Sodhi for your insightful advice and guidance through our weekly meetings, which I always look forward to. Thank you Michael Dille for helping connect my research to tremendous challenges in planetary exploration and for your feedback during our biweekly calls. Thank you Matthew Travers for helping me initiate this research journey. Thank you Richelle Bernazzoli and the CMU Undergraduate Research Office for their guidance and support through my undergraduate and graduate education.

Thank you Bob Bittner for helping support our testing operations at NREC. Thank you David Kohanbash for helping me find relevant resources to produce the experimental rig and for helping operate the total station during testing. Thank you Sebastian Schrer for providing the IMU used on the experimental platform. Thank you Ryan Darnley for helping create the first version of the inertial navigation system and David Neiman for helping with the SLAM with lines work.

Additionally, I would like to thank my fellow RI graduate students, including Suddhu, Allie, Allison, Wei, Eric, Akash, Monty, Ruoyang, Danny, Hans, Haowen, Josh Jaekel, and Josh Spisak for their help, support, advice, and feedback. Thank you Nora and BJ for your instantaneous email responses and check in meetings.

I would like to thank my parents, Howard and Simone, my sister, Jacqueline, and my partner, Rachel, for their continuous guidance and support.

Thank you Red Whittaker for your ongoing mentorship. Walking into your office during the first week of my freshman year at Carnegie Mellon inspired me to pursue a career in robotics and write this thesis.

I further acknowledge and appreciate funding support from a NASA Space Technology Graduate Research Opportunity (NSTGRO), the Department of Energy Robotics Traineeship, and the National Science Foundation. Thank you to the Barry Goldwater Scholarship Foundation for their initial support for my undergraduate education and research.

Thank you Carnegie Mellon for an unforgettable and transformative experience over the past 5 years. I will forever be grateful for being a part of this incredible community.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Scope and Approach . . . 2
- 1.3 Organization . . . 3
- 1.4 Contributions . . . 3

2 Background **5**
- 2.1 SLAM and Factor Graphs . . . 5
  - 2.1.1 Introduction to State Estimation . . . 5
  - 2.1.2 Factor Graphs . . . 6
  - 2.1.3 Performing MAP Inference . . . 8
  - 2.1.4 Front-end of SLAM . . . 8
- 2.2 Ground Penetrating Radar . . . 9
  - 2.2.1 Introduction to GPR . . . 9
  - 2.2.2 GPR in Robotics . . . 9
  - 2.2.3 Data Representations . . . 9
- 2.3 Other Radar-based Perception . . . 11

3 System Design **15**
- 3.1 SuperVision Test Platform . . . 15
  - 3.1.1 Sensor Selection . . . 15
  - 3.1.2 Antenna Placement . . . 17
- 3.2 Signal processing . . . 17
  - 3.2.1 Global and Local Features in GPR Data . . . 17
  - 3.2.2 Signal Pre-processing for GPR Localization and Mapping . . . 19
- 3.3 CMU-GPR Dataset . . . 20
  - 3.3.1 Data . . . 21
  - 3.3.2 Development Tools . . . 21
  - 3.3.3 Trajectory Data Types . . . 23
  - 3.3.4 Relevance to the Robotics Community . . . 23

4 GPR-based Localization **25**
- 4.1 Introduction . . . 25
- 4.2 Background and Related Work . . . 26

4.2.1 Prior Work on Localizing GPR (LGPR) . . . 26
4.2.2 IMU Preintegration . . . 29
4.2.3 Learned Sensor Models . . . 30
4.3 Learned Feature-based Localization Formulation . . . 30
4.3.1 Overview of Approach . . . 30
4.3.2 Problem Formulation . . . 31
4.4 Submap Construction . . . 32
4.5 Learned Sensor Model . . . 34
4.5.1 Feature Learning . . . 34
4.5.2 Submap Comparison . . . 35
4.5.3 Spatial Correlation . . . 35
4.5.4 Transform Prediction . . . 36
4.6 Training the Transform Prediction Network . . . 36
4.6.1 Ground Truth Positions . . . 36
4.6.2 Artificial Homography . . . 37
4.7 Factor Graph Optimization . . . 38
4.7.1 GPR Factor . . . 38
4.7.2 IMU Preintegration . . . 38
4.7.3 Wheel Encoder Factor . . . 38
4.7.4 Optimization . . . 39
4.8 Experimental Results . . . 39
4.8.1 GPR Factor Learning . . . 39
4.8.2 Factor Graph Optimization . . . 39
4.8.3 Qualitative Evaluation . . . 40
4.8.4 Quantitative Evaluation . . . 40
4.8.5 Optimizer Runtime . . . 41

5 GPR-based Localization and Mapping **43**
5.1 Introduction . . . 43
5.2 Background and Related Work . . . 44
5.2.1 Line Parameters in $\mathbb{R}^2$ . . . 44
5.2.2 Line Parameters in $\mathbb{R}^3$ . . . 44
5.2.3 Hyperbola Detection in GPR Data . . . 44
5.3 GPR Simultaneous Localization and Mapping Formulation . . . 45
5.3.1 Overview of Approach . . . 45
5.3.2 Problem Formulation . . . 46
5.4 Line Measurement Model . . . 47
5.5 Line Geometry . . . 48
5.6 Factor Graph Optimization . . . 49
5.7 Experimental Results . . . 51
5.7.1 Overview of Methods . . . 51
5.7.2 Simulated Experiments . . . 51
5.7.3 Real-World Results . . . 53

6 Conclusion **57**
6.1 Contributions 57
6.2 Observations and Future Work 58

Bibliography **61**

List of Figures

1.1 Instances where GPR-based positioning is most effective: **(a)** place recognition, **(b)** semi-static environments, **(c)** occlusion, **(d)** challenging weather, **(e)** sparsely featured environments. 1

1.2 Examples of visually degraded settings where robots must perform robustly, which include search and rescue [106], planetary exploration [101], and self-driving vehicles [29]. 2

1.3 Single channel GPR data depicting subsurface stratification and an underground pipe. 3

2.1 Example of simultaneous localization and mapping in an office environment from [76]. The estimated trajectory of the system is shown as blue keyframes. The map of the environment is shown as colorized points. 6

2.2 Example factor graph based on [30]. 8

2.3 Components of received signal by GPR systems from [43]. *Tx* is the transmitting antenna and *Rx* is the receiving antenna. 10

2.4 A-scan obtained over a subsurface pipe. **(a)** scenario where radar is located above the top of a subsurface pipe. **(b)** resulting reflection from the pipe observed in proceed trace. The trace contains amplitude information as a function of time, which corresponds to depth. 11

2.5 Hyperbolic features in GPR data. **(a)** description of common scenario with multiple views of a subsurface pipe through antenna motion. **(b)** corresponding GPR *B-scan* image depicting two way travel time from signal propagation in spatial domain. Visualization based on [46]. 12

2.6 Sparse three-dimensional representations for GPR data retrieved from [46]. **(a)** transparency-based volume visualization. **(b)** threshold-based visualization. . . . 13

3.1 Experimental setup for GPR subsurface perception datasets. 16

3.2 Data logging setup for SuperVision testing. 16

3.3 **(a)** GPR image constructed with 2.5cm antenna-ground separation. **(b)** GPR image constructed with ground-coupled antenna configuration. 17

3.4 Global features present in GPR data. In this example, the robot moves from soil to sidewalk, causing a change in general banding features in the B-scan. 18

3.5 GPR radargram containing distinct texture from [6]. 18

3.6 Overview of GPR image construction. **(a)** Unprocessed localized traces received by the device. **(b)** Horizontally stacked traces from (a), where the amplitudes correspond to pixel intensity. **(c)** Measurements after filtering and gain. **(d)** Final image after thresholding. 19
3.7 Experimental rig data acquisition in different environments. 20
3.8 Testing environments for CMU-GPR dataset. 21
3.9 **(a)** Directory layout for the CMU-GPR dataset. **(b)** Data format by sensor measurement type. 22
3.10 API for constructing processed images from CMU-GPR dataset. 23

4.1 Concept diagram for localizing GPR system described in [24]. 26
4.2 Particle swarm optimization registration approach for LGPR system described in [24]. 27
4.3 Correlation of test run and LGPR map in different weather conditions from [80]. 28
4.4 Error comparison from ground truth in varying weather conditions from [80]. . . 28
4.5 Estimating poses for a ground vehicle using subsurface measurements from Ground Penetrating Radar (GPR) as inference over a factor graph. Since GPR measurements are challenging to correlate, a relative transformation is learned between submaps to correct the system's pose. 31
4.6 Overview of the Ground Encoding system. On the left, two submaps containing similar subsurface structures are acquired during forward-backward trajectory that has accumulated drift. An autoencoder maps these submaps to feature activation maps, which are used by the transform prediction network to find the relative transformation. Factors denoting the measurement likelihoods from GPR, inertial, and wheel encoder measurements are incorporated into the graph to estimate the system's latent states $x_t = [s_t, v_t, b_t]^T \; \forall t \in \{1 \; ... \; T\}$. The trajectory on the right shows the outcome of the GPR correction. 33
4.7 Forward-backward trajectory from Figure 4.6 broken down into submaps. 33
4.8 Autoencoder architecture used for feature learning. A raw GPR radargram is provided to a ResNet-18 autoencoder, which reduces the image to a series of feature activations. A second ResNet decodes these activation maps to reconstruct the original image. 34
4.9 Summary of spatial correlation layer for submap transform prediction. Feature activation maps are provided to a spatial correlation function, which computes a curve of image correlations with translation. The maximum values of each curve are compiled into a correlation vector. 35
4.10 Actual versus predicted transformation **(a)** without spatial correlation and **(b)** with spatial correlation. The spatial correlation layer converts it into a linear relationship, simplifying the learning process for the transform prediction network. 36
4.11 Examples of transformations from green coordinate frame to red coordinate frame identified using ground truth data. 37
4.12 Sliding frames from odometry integration depicting synthetic motion. Frames contain the original submap image, reconstruction from the autoencoder, reconstruction loss, and a subset of activation maps. 37

4.13 Qualitative evaluation of vehicle trajectory estimate over time. **(a)** The robot follows a trajectory that loops over itself twice. **(b)** The robot starts moving on the vertical segment, then moves along the horizontal segment, before finally revisiting the vertical segment. In both cases, the learned sensor model follows the ground truth data better than the engineered method despite receiving the same loop closure detections. 41
4.14 RMSE Absolute Trajectory Error (ATE) for each testing environment. 41
4.15 RMSE ATE over time. **(a)** Corresponds to the trajectory from Fig. 4.13(a). **(b)** Corresponds to the trajectory from Fig. 4.13(b). In both cases, loop closure events reduce the accumulation of odometry error. 42
4.16 Optimizer runtime for entire duration of *nsh_b* trials. 42

5.1 Image depicting straight line underground pipes from [82]. 43
5.2 Automated hyperbola detection in GPR B-scan from [32]. **(a)** post-processed GPR B-scan. **(b)** Masked image containing hyperbola. **(c)** Hyperbolic curves overlaid on B-scan. 45
5.3 Trajectory of robot observing continuous linear feature. The red lines are GPR submaps. In each submap, a prominent hyperbola feature is found, representing an observation of the linear landmark. 45
5.4 Graph of robot states and multiple observations of a single line-feature. 46
5.5 Forward distance measurement from robot to feature in GPR submap. 47
5.6 Process to compute distance from robot to linear feature as a function of robot pose and line parameters. 48
5.7 Recipe for determining initial values for line parameters in optimization. 50
5.8 Qualitative comparison of simulated trajectories. Localization and SLAM approaches in **(a)** and **(b)** improve the trajectory estimate. *SLAM without re-visitation (1 line)* is not shown for figure clarity. 52
5.9 Qualitative comparison of estimated line features from trajectories in Figure 5.8. In **(a)** and **(b)**, the line parameters are well-estimated. The line with re-visitation approach yields improved performance since the angle parameter is initialized closer to the true value. 52
5.10 RMSE Absolute Trajectory Error (ATE) for different GPR-based positioning model with simulated data. 54
5.11 RMSE of line of estimated Hough line parameters for simulated data. 54
5.12 Qualitative comparison of SLAM approaches for GPR-based positioning in real-world environment. **(a)** two images of the test environment. **(b)** estimated trajectories using GPR-based SLAM methods. **(c)** estimated features from GPR-based SLAM methods. 55

6.1 Different air-coupled and multi-channel GPR systems from [1, 2, 80] respectively. 57
6.2 Data from GROUNDED localizing GPR dataset. Retrieved from [81]. 59

List of Tables

3.1 Filter sequence for GPR signal pre-processing. 20

4.1 Validation losses for different GPR models (cm) 40

5.1 Trajectory and landmark error for real-world testing 56

Chapter 1

Introduction

1.1 Motivation

(a) (b) (c) (d) (e)

Figure 1.1: Instances where GPR-based positioning is most effective: **(a)** place recognition, **(b)** semi-static environments, **(c)** occlusion, **(d)** challenging weather, **(e)** sparsely featured environments.

Ground penetrating radar (GPR) is a electromagnetic tool used for underground inspection, but it has the potential to be used for localization and mapping. Unlike traditional robotics sensing modalities, like LIDAR and cameras, GPR performs robustly in challenging, visually degraded environments, and is mostly invariant to spatio-temporal changes. Examples of visually degraded environments are shown in Figure 1.2, which include dirt, dust, and inclement weather. Robots must operate robustly to perform critical tasks and often are not accessible to repair. These challenges require fault-tolerant sensing, motivating the use of GPR for localization and mapping. Figure 1.1 depicts common robotics challenges where GPR-based positioning can perform effectively.

GPR performs robustly in visually degraded environments, such as fog, snow, rain, and dust. Prior work by Ort et al. has effectively used GPR for localization for autonomous navigation in varying inclement weather conditions [80]. During this work, GPR and GPS together enabled robust lane tracking during different challenging weather scenarios.

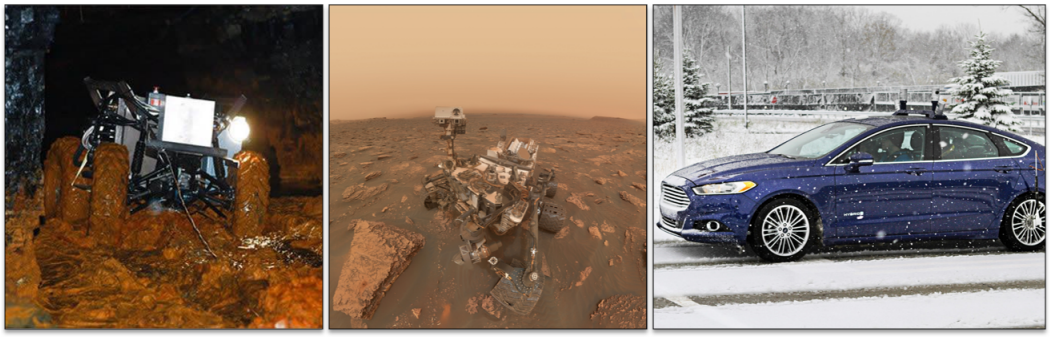

Figure 1.2: Examples of visually degraded settings where robots must perform robustly, which include search and rescue [106], planetary exploration [101], and self-driving vehicles [29].

GPR measurements are more invariant to spatio-temporal change than traditional sensor measurements since subsurface features are often consistent over time. Suppose that a robot is operating in a subsurface mine where the line of sight environment may change due to mining operations. However, subsurface features should remain mostly unchanged, making GPR a relevant sensor for vehicle positioning.

While the advantages of GPR as a positioning method are prevalent, GPR-based perception is an unsolved challenge. GPR data is notoriously difficult to interpret since image generation has a complex dependency on subsurface composition and radar physics, which itself varies with sensors and variations in subsurface electromagnetic properties. As opposed to cameras and LIDAR, which provide very rich context about a robot's position in an environment, GPR measurements only provide information about the subsurface immediately around the robot. These sensor benefits and limitations motivate our research on identifying representations to meaningfully use subsurface information for localization and mapping.

1.2 Scope and Approach

In this thesis, we describe different methods for localization and mapping using GPR sensors. We develop these methods using a low cost, off-the-shelf, single channel GPR system.

The first part of this thesis discusses the design of a platform to acquire and process GPR data, specifically for localization and mapping. This system and GPR data is shown in Figure 1.3.

The second part of this thesis–on localization–explores a novel approach for place recognition through comparison of GPR submaps. We describe the shortcomings of possible representations (single traces, point clouds, unprocessed images) and different sensor models to estimate a robot's trajectory. We show our results with actual GPR data collected in three GPS-denied environments with the experimental test rig discussed in the first part.

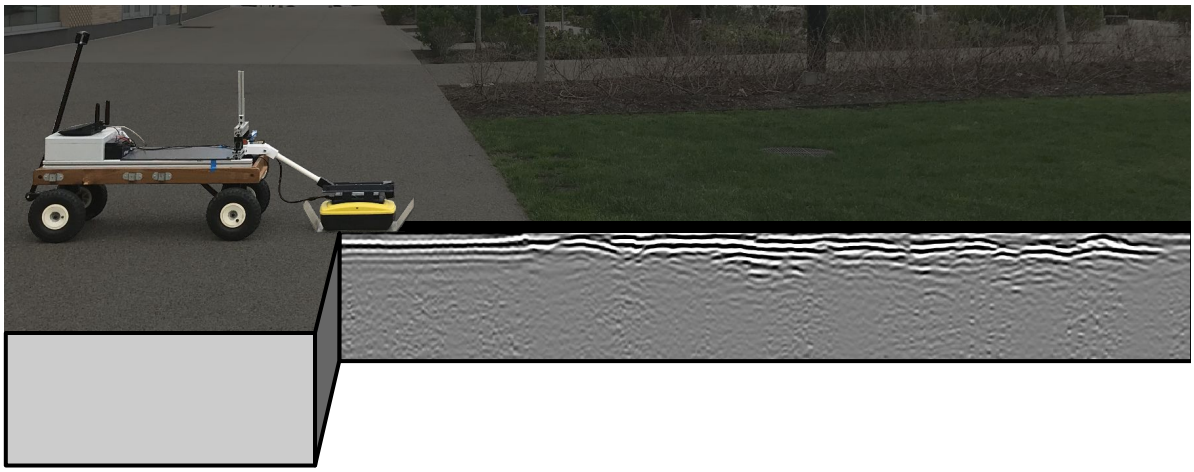

Figure 1.3: Single channel GPR data depicting subsurface stratification and an underground pipe.

The third part of this thesis–on simultaneous localization and mapping–leverages common linear and continuous underground structures to enable simultaneous localization and mapping using GPR sensors. This method relaxes the requirement to revisit previous positions or to maintain a prior map of the environment, a common limitation of GPR-based positioning systems. Linear features are common in the real-world, often taking the form of subsurface pipes and geologic fractures [92, 109]. We demonstrate our method using simulated and real-world data.

1.3 Organization

This thesis is organized as follows. In Chapter 2, we cover background theory on SLAM, factor graphs, ground penetrating radar, and signal processing. In Chapter 3, we describe the design of a GPR localization and mapping platform, consisting of an experimental rig and signal pre-processing approach. In Chapter 4, we describe the theory and implementation of our GPR localization method. In Chapter 5, we describe a GPR-based simultaneous localization and mapping (SLAM) system. In Chapter 6, we conclude this thesis and propose future work to enable robust, real-world deployment of GPR-based positioning systems.

1.4 Contributions

In Chapter 3, we describe the design of a system to collect and process GPR data. Our main contributions are:

1. An experimental rig for GPR localization and mapping data collection.
2. A signal processing routine to preserve underground features needed for localization and mapping.

3. An open source dataset for evaluating GPR localization and mapping techniques.

In Chapter 4, we propose a localization formulation based on GPR submaps and a learned sensor model. To the best of our knowledge, this is the first GPR-based localization technique relying on smoothing-based methods. Our main contributions are:

1. A formulation of the GPR localization problem as inference over a factor graph without a prior map.
2. A learnable GPR sensor model based on submaps.
3. Experimental evaluation on a test platform with a single-channel GPR system operating in three different GPS-denied environments.

In Chapter 5, we present a full SLAM system that jointly infers robot states and underground line features. To the best of our knowledge, this is the first formulation for SLAM using subsurface features. Here, our contributions are:

1. A method to discretely model line features in a GPR radargram for positioning.
2. A SLAM framework that estimates line parameters over multiple robot observations.
3. Results from simulated and real-world experimentation.

Chapter 2

Background

In this chapter, we introduce the topics of simultaneous localization and mapping (SLAM) and ground penetrating radar (GPR). We begin by providing an overview of the traditional SLAM problem. Then, we introduce factor graphs for solving probabilistic inference problems. We then describe how GPR systems operate and some common representations for GPR data.

2.1 SLAM and Factor Graphs

2.1.1 Introduction to State Estimation

When robots do not have access to ground truth position and orientation information, they must estimate these critical parameters from noisy sensor data. Sensors can provide proprioceptive (e.g. measurements from a wheel encoder or inertial sensor) and exterioceptive information (e.g. capture relative positions with respect to a map) [75]. Proprioceptive sensors drift over time since integration of noisy measurements causes error to accumulate. In these cases, we wish to use external information, such as a scan of the robot's surroundings, to correct for this error.

In some applications, a robot must localize within a known map of an environment. In a traditional localization problem, a robot can know the location of a set of landmarks and use this information for trajectory estimation [18]. However, we often do not have *a priori* knowledge of a robot's operating environment, requiring us to construct a model of the robot's environment. This process is known as mapping. The problem of estimating a robot's pose while also building a map of the surrounding environment is known as *simultaneous localization and mapping*, abbreviated as SLAM [102].

A traditional SLAM problem is visualized in Figure 2.1. In this example, a camera is moved around an unknown office environment. A computer extracts features from an image and as the camera moves, triangulates their 3-dimensional position. While the robot creates a map of the environment, it is able to localize itself within the environment. When a robot re-observes some region within a map, a loop closure occurs, which can be leveraged in SLAM to correct for accumulated drift.

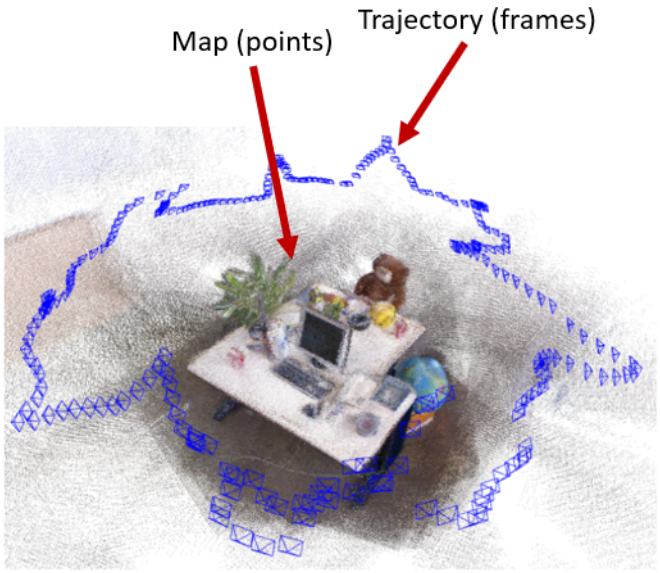

Figure 2.1: Example of simultaneous localization and mapping in an office environment from [76]. The estimated trajectory of the system is shown as blue keyframes. The map of the environment is shown as colorized points.

In the SLAM problem, we estimate the joint density of the robot's states ($\mathcal{X}$) and map consisting of landmark sites ($\mathcal{L}$) conditioned on sensor measurements (Z) [61, 72]. We define the robot's states over time as $\mathcal{X} = \{\mathbf{x}_t\}_{t=1}^{T}$, landmark locations as $\mathcal{L} = \{\mathbf{l}_m\}_{m=1}^{M}$, and robot's sensor measurements as $\mathcal{Z} = \{\mathbf{z}_n\}_{n=1}^{N}$. We wish to infer robot's states and landmark locations that maximize the conditional density $p(\mathcal{X}, \mathcal{L}|\mathcal{Z})$. Therefore, the optimal solution to the SLAM is the *maximum a posteriori* (MAP) estimate defined as:

$$\hat{\mathcal{X}}, \hat{\mathcal{L}} = \underset{\mathcal{X},\mathcal{L}}{\operatorname{argmax}}\, p(\mathcal{X}, \mathcal{L}|\mathcal{Z}) \tag{2.1}$$

$$\propto \underset{\mathcal{X},\mathcal{L}}{\operatorname{argmax}}\, p(\mathcal{Z}|\mathcal{X}, \mathcal{L})p(\mathcal{X}, \mathcal{L}) \tag{2.2}$$

2.1.2 Factor Graphs

Classical positioning systems often use filtering-based approaches, such as the Extended Kalman Filter (EKF) [58, 96]. Filtering-based approaches are commonly used today, but can suffer from high computational cost and accumulated linearization errors [99]. In order to mitigate some of these challenges, we use smoothing-based methods, which have been proven to be more accurate and efficient than filtering-based approaches [18].

A factor graph is a bipartite graph with two types of nodes: variables $x \in \mathcal{X}$ and factors $\phi(\cdot) : \mathcal{X} \to \mathbb{R}$. Variable nodes are the latent states to be estimated, and factor nodes encode constraints on these variables, such as measurement likelihood functions.

The smoothing problem is often solved using MAP inference. While MAP inference is known to be an NP-Hard problem [94], conditional independence assumptions between associated variables enables allows us to represent the joint probability as:

$$p(\mathcal{X}, \mathcal{L}, \mathcal{Z}) = p(\mathbf{x}_0) \prod_{n=1}^{N} p(\mathbf{z}_n | \mathbf{x}_{\alpha_n}, l_{\beta_n}) \prod_{t=1}^{T} p(\mathbf{z}_t | \mathbf{x}_{t-1}, \mathbf{x}_t) \tag{2.3}$$

$$\log p(\mathcal{X}, \mathcal{L}, \mathcal{Z}) = \log p(\mathbf{x}_0) + \sum_{n=1}^{N} \log p(\mathbf{z}_n | \mathbf{x}_{\alpha_n}, l_{\beta_n}) + \sum_{t=1}^{T} \log p(\mathbf{z}_t | \mathbf{x}_{t-1}, \mathbf{x}_t) \tag{2.4}$$

Typically we assume that the data association $\mathcal{D} = \{(\alpha_n, \beta_n)\}_{n=1}^{N}$ of robot poses $\mathbf{x}_{\alpha_n}$ and landmarks l_{β_n} is known *a priori*. Additionally, we often make a simplifying assumption that measurements are drawn from a zero-mean Gaussian distribution:

$$\mathbf{z}_0 = \mathbf{x}_0 + \nu_0 \tag{2.5}$$

$$\mathbf{z}_n = h(\mathbf{x}_{\alpha_n}, l_{\beta_n}) + \nu_n \tag{2.6}$$

$$\mathbf{z}_t = h(\mathbf{x}_{t-1}, \mathbf{x}_t) + \nu_t \tag{2.7}$$

where, ν_0, ν_n, and ν_t are noise terms drawn from a Gaussian distribution with covariance Σ_0, Σ_n, and Σ_t respectively. $\mathbf{z}_0$ is the prior term that constrains the first pose in the optimization. This term is often given small Σ_0 to ground the optimization. h is a measurement function that maps latent variables to a measurement prediction. Since we are estimating the values of the latent variables that maximize the joint likelihood, we can remove constant terms in the factorization and write the individual densities as:

$$p(\mathbf{x}_0) \propto \exp\{-\frac{1}{2}||\mathbf{z}_0 - \mathbf{x}_0||^2_{\Sigma_0}\} \tag{2.8}$$

$$p(\mathbf{z}_n | \mathbf{x}_{\alpha_n}, l_{\beta_n}) \propto \exp\{-\frac{1}{2}||\mathbf{z}_n - h_n(\mathbf{x}_{\alpha_n}, l_{\beta_n})||^2_{\Sigma_n}\} \tag{2.9}$$

$$p(\mathbf{z}_t | \mathbf{x}_{t-1}, \mathbf{x}_t) \propto \exp\{-\frac{1}{2}||\mathbf{z}_t - h(\mathbf{x}_{t-1}, \mathbf{x}_t)||^2_{\Sigma_t}\} \tag{2.10}$$

By manipulating the MAP objective, we show that maximizing the posterior probability proportional to the negative log probability of the joint probability:

$$\hat{\mathcal{X}}, \hat{\mathcal{L}} = \underset{\mathcal{X}, \mathcal{L}}{\operatorname{argmax}}\, p(\mathcal{X}, \mathcal{L} | \mathcal{Z}) \tag{2.11}$$

$$\propto \underset{\mathcal{X}, \mathcal{L}}{\operatorname{argmax}}\, p(\mathcal{X}, \mathcal{L}, \mathcal{Z}) \tag{2.12}$$

$$= -\underset{\mathcal{X}, \mathcal{L}}{\operatorname{argmin}}\, \log p(\mathcal{X}, \mathcal{L}, \mathcal{Z}) \tag{2.13}$$

Given the objective defined in Equation 2.13, we can write joint density using the factorized measurement distributions as:

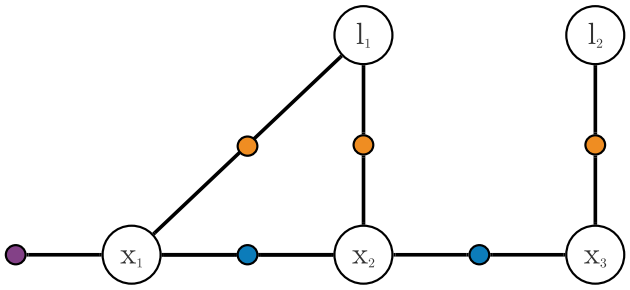

Figure 2.2: Example factor graph based on [30].

$$\hat{\mathcal{X}}, \hat{\mathcal{L}} = \underset{\mathcal{X},\mathcal{L}}{\operatorname{argmin}}(||\mathbf{z}_0 - \mathbf{x}_0||^2_{\Sigma_0} + \sum_{n=1}^{N} ||\mathbf{z}_n - h_n(\mathbf{x}_{\alpha_n}, l_{\beta_n})||^2_{\Sigma_n} + \sum_{t=1}^{T} ||\mathbf{z}_t - h(\mathbf{x}_{t-1}, \mathbf{x}_t)||^2_{\Sigma_t}) \tag{2.14}$$

This objective can be written more generally by combining all measurements and latent variables into a single state vector $\mathcal{X}$:

$$\hat{\mathcal{X}} = \underset{\mathcal{X}}{\operatorname{argmin}} \sum_{i=1}^{N} ||\mathbf{z}_i - h_i(\mathbf{X})||^2_{\Sigma_i} \tag{2.15}$$

2.1.3 Performing MAP Inference

Since the measurement function is often nonlinear, the MAP objective can be solved using nonlinear least squares optimization. These problems cannot be solved directly and require an initial estimate and an iterative solution [30]. A variety of algorithmic approaches have been used to solve this problem, including steepest decent, Gauss-Newton (GN), Levenberg-Marquardt, and Powell's Dogleg [30].

In addition to these classical optimization approaches, incremental solvers, such as iSAM2, can be used for real-time estimation, which is appropriate for the incremental nature of SLAM [57].

2.1.4 Front-end of SLAM

The graphical back-end of the SLAM system often assumes that the graph structure is known *a priori* by the front-end. In order to effectively reason about the world, a robot must convert raw sensor data into structured representations. The front-end is responsible for data preparation,

modeling, and association to extract spatial relations between individual observations.

The front-end varies for different sensor modalities. For LIDAR-based SLAM systems, scan alignment is a tool to identify odometry from related scans [114]. Visual SLAM systems utilizing monocular or stereo cameras often rely on sparse *feature-based* representations [71, 76]. Prior work in RGB-D-based SLAM systems often use *direct methods* that utilize all pixels in the input frame in the estimation problem [78].

2.2 Ground Penetrating Radar

2.2.1 Introduction to GPR

GPR is a tool for subsurface imaging that uses radio waves to probe low loss dielectric materials [3]. GPR systems transmit electromagnetic (EM) waves and receive scattered reflections from interfaces of different dielectric properties. These interfaces often have several different classifications, which include:

1. Boundary: stratification and layering of the subsurface.
2. Point: a defined object with limited extent.
3. Line: often continuous structures, such as pipes and fractures.

GPR systems measure the two way travel time of the signal transmitted and received by the antenna [28]. Physical modeling of GPR systems of varying topology have been extensively reviewed in [3, 27, 95]. Applications of GPR systems in different domains have been described in great depth in [14, 103].

2.2.2 GPR in Robotics

Most use of GPR in the robotics domain has been for passive inspection, which includes excavation of subsurface objects [47], planetary exploration [40, 45, 63], mine detection [79, 91], bridge deck inspection [62], and crevasse identification [107].

2.2.3 Data Representations

1D A-Scans

An *A-scan*, otherwise known as a 1D trace, is formed by measuring the amplitude response of a scattered EM wave. Since the subsurface is lossy, the received signal attenuates as a function of depth, conductivity, permittivity, and permeability [3]. In order to effectively compensate for this loss and produce an interpretable image, a gain function is applied to amplify relevant features.

A trace contains three noticeable components: the air wave, the ground wave, and the non-line-of-sight reflected signal. The air wave is the observed signal directly after transmission. The

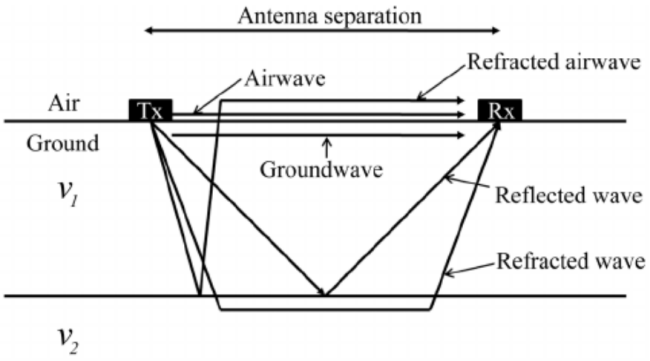

Figure 2.3: Components of received signal by GPR systems from [43]. *Tx* is the transmitting antenna and *Rx* is the receiving antenna.

ground wave is the reflection from the air-ground interface. In ground-coupled antennas, this may not be visible. This characteristic is most noticeable with air-coupled antennas, where there is an air gap between the antenna and the ground [31, 43]. The remaining component of the trace is a combination of the reflected signal and noise. The ability to detect a relevant feature is defined by the radar system's sensitivity, which can be measured by the system's signal-to-noise ratio (SNR), constant false alarm rate (CFAR), and other metrics [95]. These different components to the signal are shown in Figure 2.3. An example of an *A-scan* is shown in Figure 2.4.

Traces can be transformed from a timescale to a depth scale based on the signal velocity in the subsurface. This value can obtained by performing a velocity calibration, which can be accomplished by fitting a hyperbola to a feature observation or by specifically defining the depth based on assumptions or known information. In most cases, a single velocity model is used to characterize depth. In cases where stratification is more present, a multi-velocity model can be implemented.

2D B-Scans

A *B-scan*, otherwise known as a brightness scan, is a collection of equally spaced, one-dimensional traces, formatted as a two-dimensional image. Features, like pipes, appear in a *B-scan* as hyperbolas. These features occur since the radiation emitted by the antenna often follows a conical shape, causing the signal to propagate outward to features around the antenna. As the sensor gets loser to the feature, the depth of the reflection should decrease. Given this physical context, we often assume that the feature's location is co-located with the top of a hyperbola. Figure 2.5 conveys how these hyperbolic features appear in the data.

Volumetric Information

GPR data can also be represented as a three-dimensional volume. There are two ways to construct this representation: an multi-channel sensor or a grid scan with a single-channel sensor. In both

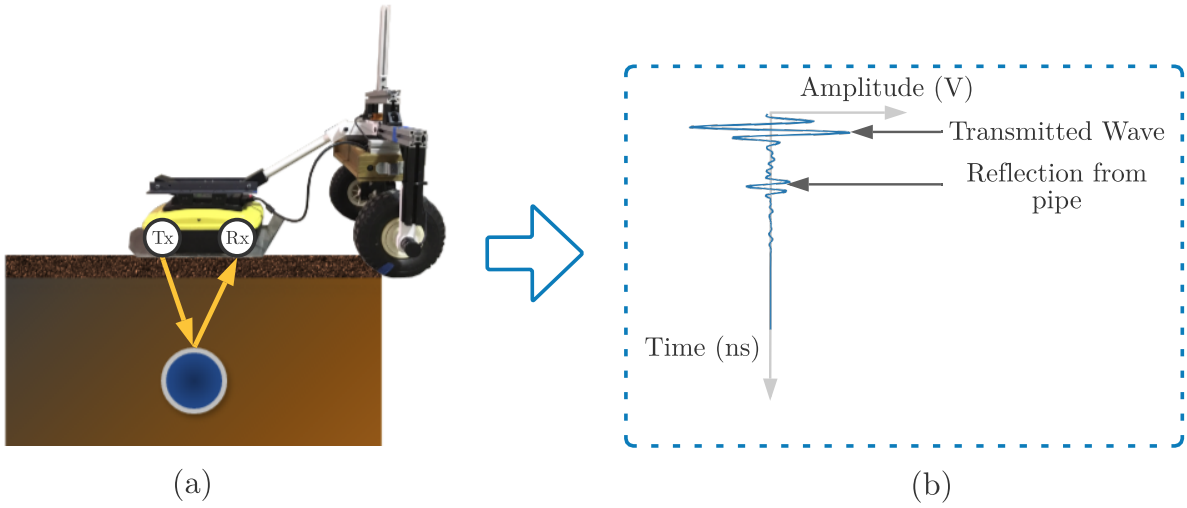

Figure 2.4: A-scan obtained over a subsurface pipe. **(a)** scenario where radar is located above the top of a subsurface pipe. **(b)** resulting reflection from the pipe observed in proceed trace. The trace contains amplitude information as a function of time, which corresponds to depth.

approaches, signals are interpolated to construct the volume.

It is typically unfeasible for human interpreters to perceive volumetric data. Thus, different processing techniques are used to simplify the presentation of volumetric information. These include:

1. Slicing the volume to view relevant B-scans or horizontal layers [35].
2. Making the voxel transparency inversely proportional to the signal amplitude [46]. One limitation of this approach is that if the volume is large, occlusion will remain and it will be difficult to visualize prevalent features. An example of this representation is shown in Figure 2.6(a).
3. Applying a minimum amplitude threshold [46]. This approach risks removing potentially important lower amplitude features. Since the threshold is often characteristic of the environmental condition and signal processing, a user must empirically determine the best value that visualizes the desired features. An example of this representation is shown in Figure 2.6(b).

2.3 Other Radar-based Perception

Prior Work

Radar-based perception has been shown to perform more robustly than conventional spatial or visual sensors in inclement weather [4, 12, 70, 83, 93]. Work in this space typically involves a

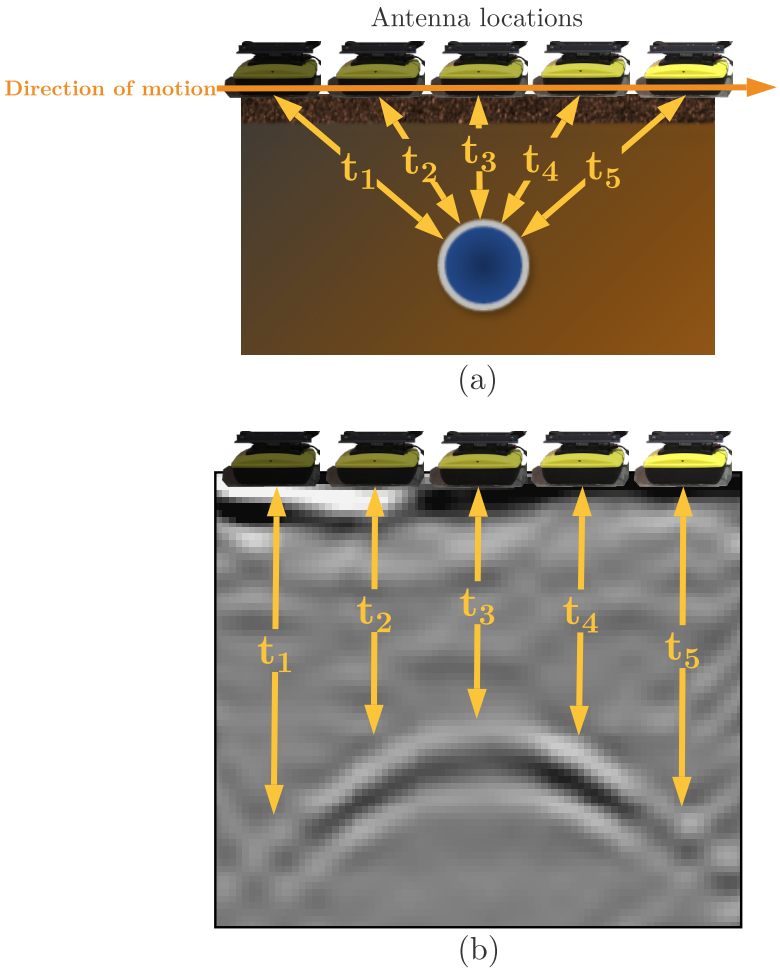

Figure 2.5: Hyperbolic features in GPR data. **(a)** description of common scenario with multiple views of a subsurface pipe through antenna motion. **(b)** corresponding GPR *B-scan* image depicting two way travel time from signal propagation in spatial domain. Visualization based on [46].

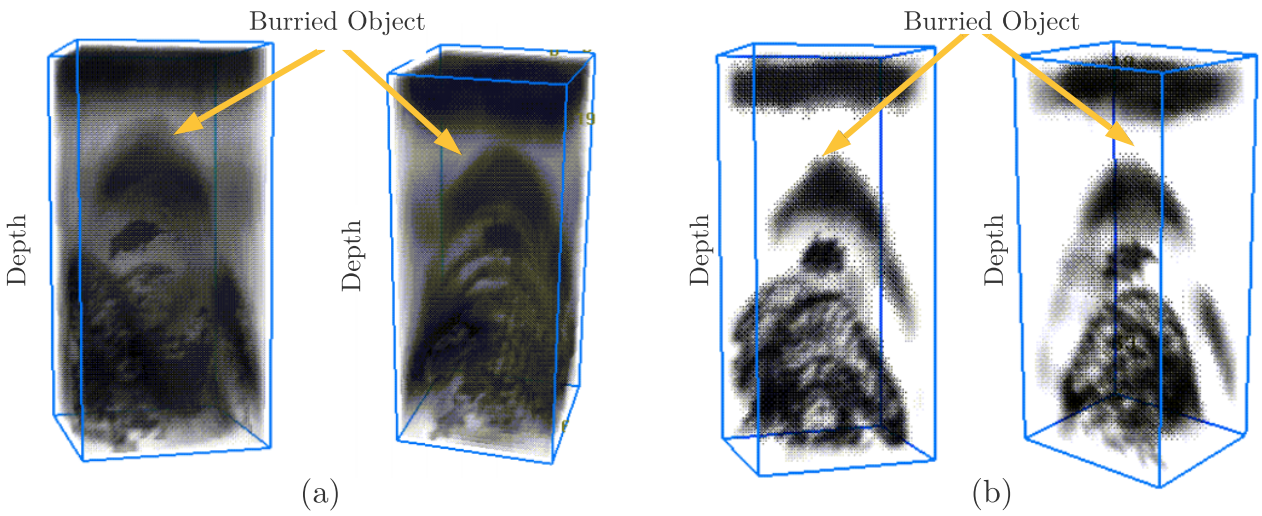

Figure 2.6: Sparse three-dimensional representations for GPR data retrieved from [46]. **(a)** transparency-based volume visualization. **(b)** threshold-based visualization.

rotating millimeter-wave FMCW scanning radar [12, 93] or fixed array radar relying on beamforming [13, 49]. These radar systems are intended to mimic LIDAR and capture surface-level geometry. Radar data is often more noisy than LIDAR, requiring additional post-processing to extract meaningful information [10, 105]. This information has been used for egomotion estimation [4, 10, 11, 21, 22, 85], place recognition [41], re-localization [88], localization within satellite images [100], registration to LIDAR point clouds [84], and semantic segmentation [59].

Comparison with GPR

Similar to traditional automotive or scanning radar applications, GPR also has good performance in challenging weather conditions. The primary difference between these different radar sensing modalities is that traditional scanning radar systems provide view of the surrounding environment around the robot while GPR simply looks at the region directly below the robot. GPR data is limited since odometry information between sequential GPR measurements is difficult to interpret and would not be as accurate as scan alignment from scanning radar systems.

We believe that GPR is most effective for long-term navigation, especially in cases where the surface environment changes. This is because subsurface features are often more stable than surface features. Thus, GPR is particularly suitable for positioning in previously mapped environments or environments with rich subsurface features, but sparse, repetitive, and / or changing surface features.

Chapter 3

System Design

In this chapter, we describe the design of a GPR system for collecting and processing data for GPR-based localization and mapping. In Section 3.1, we describe the SuperVision platorm, a custom experimental test rig produced for data collection and experimentation. In Section 3.2, we describe features in GPR data and a signal pre-processing approach designed for localization and mapping. In Section 3.3, we describe the CMU-GPR dataset, which consists of relevant navigation data along with GPR signal pre-processing utility functions for the robotics research community.

3.1 SuperVision Test Platform

3.1.1 Sensor Selection

We constructed a manually-pulled test rig named SuperVision shown in Fig. 3.1 for data acquisition. SuperVision uses a quad-core Intel NUC for compute and wirelessly transmits onboard data from an XSENS MTI-30 9-axis Inertial Measurement Unit, YUMO quadrature encoder with 1024 PPR, and a Sensors and Software Noggin 500 GPR. Readings from the IMU magnetometer were excluded due to intermittent magnetic interference commonly found in indoor environments. Ground truth data was acquired by a Leica TS15 total station. The base station logs measurements from the onboard computer and the total station to ensure consistent timing. The flow of data during experimentation is summarized in Figure 3.2.

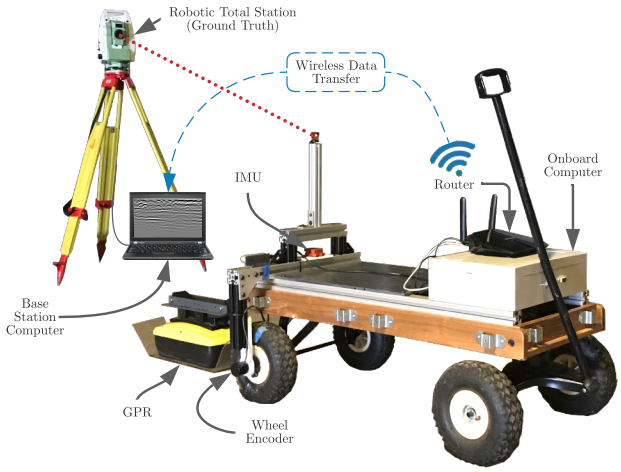

Figure 3.1: Experimental setup for GPR subsurface perception datasets.

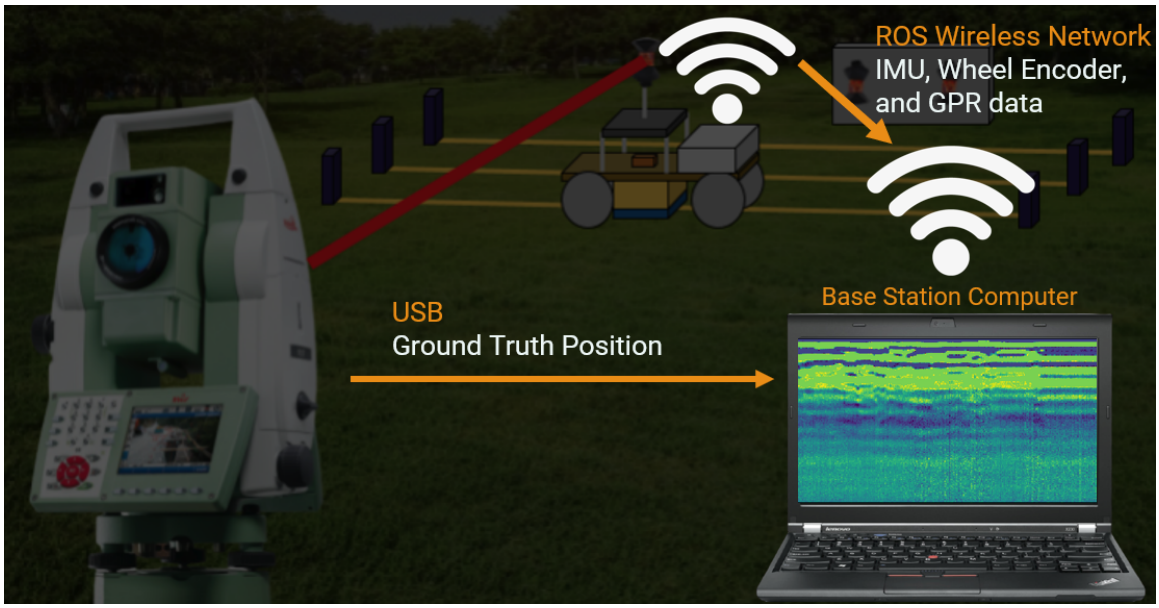

Figure 3.2: Data logging setup for SuperVision testing.

3.1.2 Antenna Placement

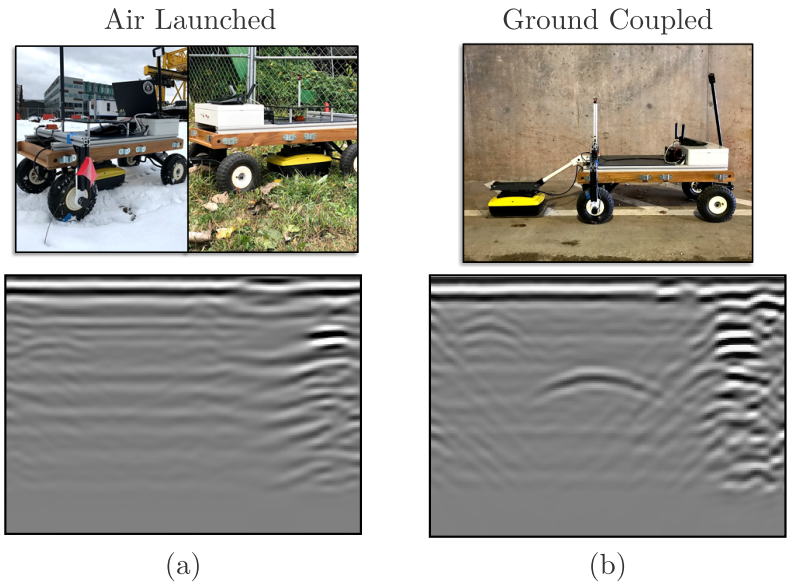

Figure 3.3: **(a)** GPR image constructed with 2.5cm antenna-ground separation. **(b)** GPR image constructed with ground-coupled antenna configuration.

In our testing, we found that GPR antenna placement was a critical design decision. Our initial design suspended the radar system 2.5cm above the ground to maintain a fixed transformation between the IMU and GPR sensor. While a fixed sensor setting like this is most common for localization systems, the small air gap between the GPR and the ground introduced ground energy losses and multi-path interactions causing poor depth penetration and repetitive ringing. To address this, we added a passive suspension to the GPR system so as to maintain constant ground contact. This improved depth penetration as shown in the right image of Fig. 3.3(c). We address methods to improve the robustness and applicability of our system in Section 6.2.

3.2 Signal processing

3.2.1 Global and Local Features in GPR Data

We define two types of features in GPR data: global features and local features. Global features define general and consistent information about an environment, such as consistent stratification. An example of these types of features is shown in Figure 3.4.

Local features capture irregular information that causes GPR scans in the same environment to be distinct and heterogeneous. Local features can include points (e.g. rocks), lines (e.g. pipes), and irregular texture or variation. An example of a radargram with substantial texture is shown in Figure 3.5.

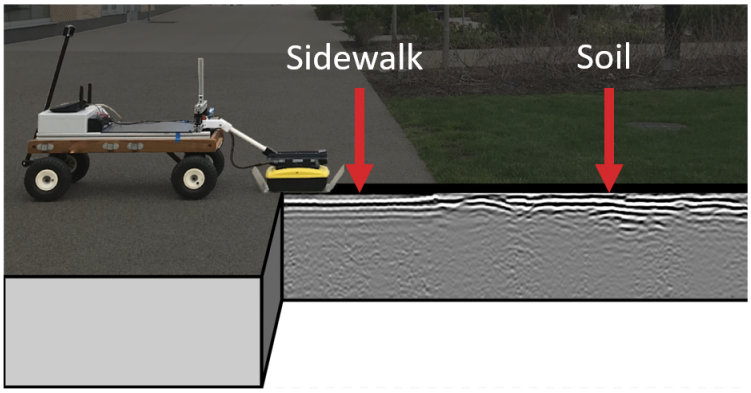

Figure 3.4: Global features present in GPR data. In this example, the robot moves from soil to sidewalk, causing a change in general banding features in the B-scan.

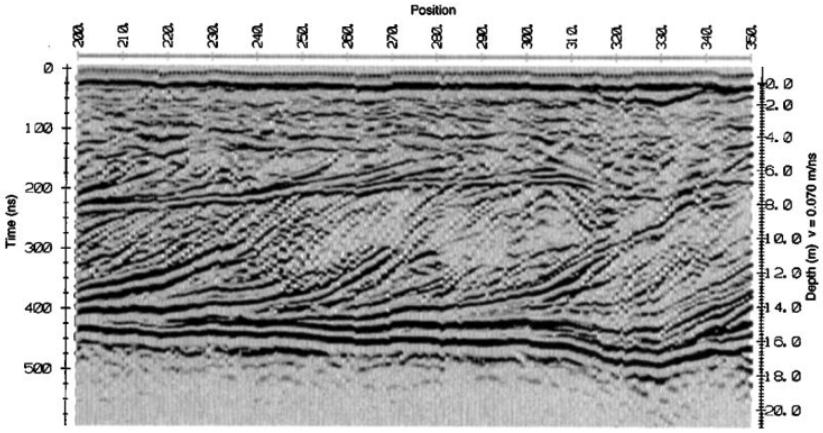

Figure 3.5: GPR radargram containing distinct texture from [6].

3.2.2 Signal Pre-processing for GPR Localization and Mapping

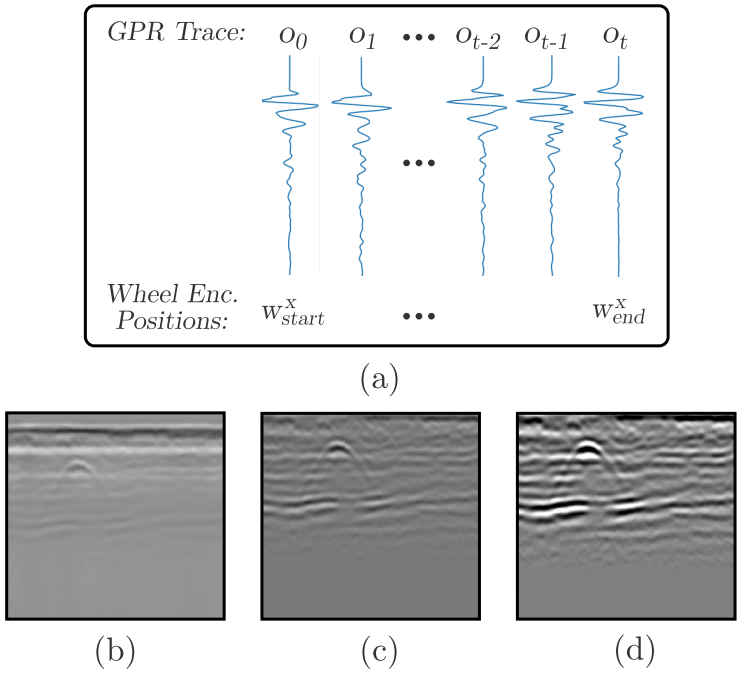

Figure 3.6: Overview of GPR image construction. **(a)** Unprocessed localized traces received by the device. **(b)** Horizontally stacked traces from (a), where the amplitudes correspond to pixel intensity. **(c)** Measurements after filtering and gain. **(d)** Final image after thresholding.

Digital signal processing techniques are needed to improve the signal-to-noise ratio of the received GPR signal, counteract signal attenuation, and remove recurrent artifacts that are intrinsic to GPR systems. Measurements o_t arrive from each position as a 1D discrete waveform called a trace, which are locally averaged to reduce noise and resampled to form a uniformly spaced image [36]. In order to increase the signal-to-noise ratio (SNR), a process called *stacking* is used. This involves averaging multiple observations in roughly the same location to reduce incoherent noise from individual traces.

Receiving signals often contain a low frequency component and DC bias caused by saturation and inductive coupling, requiring a dewow filter, which involves a DC subtraction and low-cut filter parameterized on the radar's center frequency and bandwidth [55]. Since the receiving antenna begins listening for a signal prior to transmission, we define the varying zero time of the signal as the first negative peak of the trace. We then apply a finite impulse response (FIR) filter to remove high frequency noise with less signal distortion compared to the standard bandpass filter [55]. In order to counteract attenuation, we multiply the signal by a Spreading and Exponential Compensation (SEC) gain function,

$$G = \exp(a \cdot t) \cdot t^b \tag{3.1}$$

where, a is the exponential gain constant and b is the power gain constant [50]. These parameters must be tuned for different environments since attenuation is a property of the material and

Filter Type	Application Level	Motivation
Dewow	Trace	Remove DC bias and inductive coupling
Finite Impulse Response	Trace	Reduce high frequency noise
Background Subtraction	Window	Remove repetitive ringing
SEC Gain	Trace	Amplify signal attenuation
Wavelet	Image	Reduce high frequency noise

Table 3.1: Filter sequence for GPR signal pre-processing.

Figure 3.7: Experimental rig data acquisition in different environments.

environmental condition.

GPR images have global and local context. Global features such as horizontal layers encode information about the general environment of a system, which can include boundaries like layers of asphalt in roads. Local features, which include pipes and point objects, provide salient features needed to effectively localize. To maintain prominent global features, we subtract the mean trace over all prior submaps (as opposed to over each submap) to remove repeated artifacts introduced by the radar while emphasizing local features. The signal processing process is summarized in Table 3.1 and shown in Figure 3.6.

3.3 CMU-GPR Dataset

Several open source distributions are available to process [51, 77] and simulate [104] GPR data. In this section, we describe our open-source CMU-GPR dataset, which contains SuperVision sensor data in indoor and outdoor environments as well as utility functions for GPR image construction [9]. To the best of our knowledge, this is the first open source dataset for GPR-based perception[1,2]. We are excited to hear about the release of the GROUNDED localizing GPR dataset, which will further support researchers in the GPR-based localization and mapping space [81].

[1]The CMU-GPR dataset is available at `https://github.com/rpl-cmu/CMU-GPR-Dataset`.
[2]A video describing our dataset is available at `https://youtu.be/y3XiDmFilwQ`.

Figure 3.8: Testing environments for CMU-GPR dataset.

3.3.1 Data

The CMU-GPR dataset contains short trajectories from three distinct, GPS-denied environments: a basement (*nsh_b*), a factory floor (*nsh_h*), and a parking garage (*gates_g*). These environments are shown in Figure 3.8. The dataset consists of 15 sequences where the manually-pulled experimental rig revisits previous locations, whether through forward-backward motion or by closing loops. Several distinct trajectories contain similar features, which are relevant to research on re-localization using subsurface information and further described on the project website. Additionally, we provide trajectories from outdoor environments that do not contain ground truth information, which can be used for training models.

Figure 3.9(a) shows the directory structure for a single trajectory sequence. Each zip file contains data from a sequence, where each `.csv` represents a different measurement type. The maximum size of a zipped dataset file is 4 GB. The formats for each measurement type are summarized in Figure 3.9(b).

3.3.2 Development Tools

Along with relevant datasets for localizing GPR, we provide utility code, written in Python, which processes the raw GPR data to construct images. We provide a modular routine to process one-dimensional measurements and construct an image with uniform spacing is described in Section 3.2. Additionally, we provide a script to generate submaps, which can be used for training or evaluating GPR sensor models.

Utility code to process raw GPR signals and images is provided in `signal_processing_utils.py` and used in `metric_gpr_image.py`. The implementation accepts raw, unevenly spaced GPR images and produces processed brightness scans. The base pipeline performs rubber band interpolation, mean background subtraction, dewow filtering, triangular bandpass filtering, zero time correction, SEC gain, wavelet denoising, and gaussian filtering.

```
cmu-gpr-dataset
├─ time_s-time_ns-loc-cmu-gpr
│   ├─ ts_meas.csv            # Ground truth positions
│   ├─ gpr_meas.csv           # GPR measurements
│   ├─ imu_meas.csv           # IMU measurements
│   ├─ we_odom_meas.csv       # Wheel encoder measurements
│   ├─ camera
│   │   ├─ <timestamp_s>.png  # RGB camera image
│   │   └─ ...
└─ ...
```

(a)

Measurement Type	Measurement CSV Value
GPR Measurement:	$[\ t_{stamp},\ \underbrace{amp_1, \dots, amp_{201}}_{\text{Amplitude of 1D Trace}}\ ]$
IMU Measurement:	$[\ t_{stamp},\ \underbrace{a_x, a_y, a_z}_{\text{Accelerometer } [m/s^2]},\ \underbrace{g_x, g_y, g_z}_{\text{Gyroscope } [rad/s]},\ \underbrace{w, x, y, z}_{\text{Magnetometer}}\ ]$
Wheel Encoder Measurement:	$[\ t_{stamp},\ \underbrace{dist_x}_{\text{Signed distance traversed [m]}}\ ]$
Total Station Measurement:	$[\ t_{stamp},\ \underbrace{p_x, p_y, p_z}_{\text{Ground truth position [m]}}\ ]$

(b)

Figure 3.9: **(a)** Directory layout for the CMU-GPR dataset. **(b)** Data format by sensor measurement type.

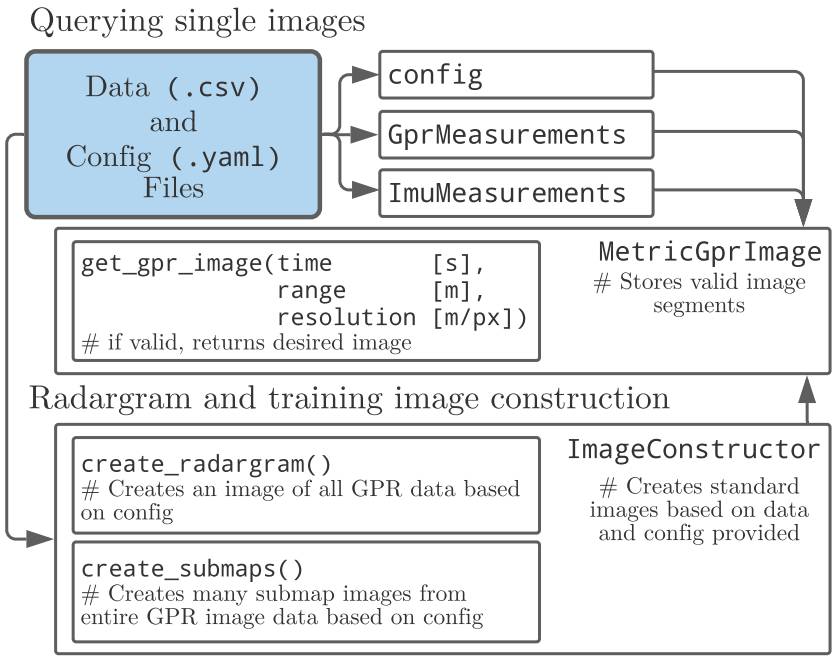

Figure 3.10: API for constructing processed images from CMU-GPR dataset.

Beyond processing the raw data, the utility code also simplifies image acquisition. The `MetricGprImage` object stores the GPR dataset and allows the client to access images based on the time of acquisition. The `ImageConstructor` object is even more abstract, allowing the client to create a traditional radargram of the entire sequence or automatically generate all valid submap images.

3.3.3 Trajectory Data Types

We primarily collected two forms of trajectories: forward-backward motion and circular loops. In both trajectory types, GPR measurements can be registered to correct accumulated drift. Data was collected in a variety of indoor and outdoor environments. Some testing environments are shown in Figure 3.7.

3.3.4 Relevance to the Robotics Community

Our motivation for providing this contribution is to encourage others in the field to take similar steps in making GPR-based perception datasets available to researchers. We believe that providing this data to the research community will spur further development of robust GPR-based localization systems for real world deployment.

Chapter 4

GPR-based Localization

4.1 Introduction

GPR-based localization can enable robust robot positioning in challenging, unstructured environments where conventional visual or spatial localization systems fail. GPR measurements are *mostly* invariant to spatio-temporal change, like different weather or changes in scene appearance, since the subsurface changes less frequently and extremely than the surface-level environment.

Consider an operational subsurface mine where continuous drilling and blasting changes the line-of-sight appearance of the scene and creates unexplored environments. These types of large-scale changes could cause visual and spatial localization systems to perform poorly. While the visual environment may change, subsurface features are typically invariant and can be used to recognize the system's location. GPR has broad relevance for localization in other unstructured environments, such as indoor areas with large crowds that would obscure a robot's field of view.

In this chapter, we will describe localizing GPR. In Section 4.2, we will discuss prior contributions in the space of localizing GPR (LGPR) as well as preliminary theory on IMU preintegration and learned sensor models. In Section 4.3, we describe our localization system named Ground Encoding. Our system models the localization problem with a smoothing objective and uses a learned sensor model to predict transformations between non-sequential GPR images. In Section 4.4, we describe GPR submap representations. In Section 4.5, we describe the methods used for the learned GPR sensor model, which include feature learning, submap comparison, spatial correlation, and transform prediction. In Section 4.6, we describe methods used to train GPR sensor models. In Section 4.7, we discuss how these transform predictions are added to the factor graph optimization. Finally, in Section 4.8, we show that the proposed learned sensor model has improved localization performance over traditional engineered feature comparison approaches used in prior work through qualitative and quantitative evaluation with real-world data.

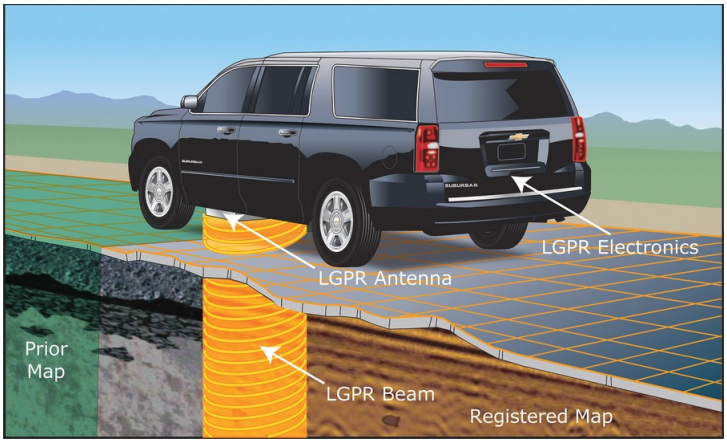

Figure 4.1: Concept diagram for localizing GPR system described in [24].

4.2 Background and Related Work

4.2.1 Prior Work on Localizing GPR (LGPR)

Research on localizing GPR originated at MIT Lincoln Laboratory, where Cornick et al. developed a custom GPR device for vehicle localization [24]. A concept of their proposed design is shown in Figure 4.1.

GPR System Design

During this work, a novel ultrawideband, stepped frequency continuous wave (SFCW), multichannel GPR sensor was designed with localization in mind [37]. The system operated at low frequencies (100-400 MHz), which was shown to be more robust, repeatable, and easier to correlate than higher frequency (e.g. 1 GHz) GPR sensors. 11-channels data channels were recorded by continuously sweeping through a 12 element array. In order to capture more dense data, the spacing between elements is 12.7cm, which is about one tenth of a center frequency wavelength. This is more narrow than conventional multichannel GPR sensors to better register scans to baseline data.

The antenna was particularly designed for lane tracking and yields polarized measurements. Meaning that the system needed to follow a nearly parallel heading to the generated map.

Localization

The LGPR system developed in [24] requires a prior map for measurement registration. First, a local grid is developed, which is queried from a database given an approximate GPS coordinate. Given the current scan, a particle swarm optimization (PSO) is used to estimate the five-dimensional state of the sensor, consisting of latitude, longitude, height, heading, and roll. The PSO compares the sweep data to the interpolated grid map using an engineered heuristic fit criterion defined by:

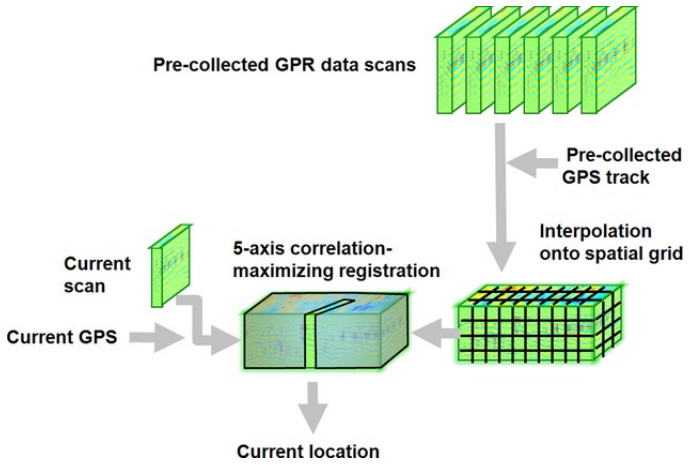

Figure 4.2: Particle swarm optimization registration approach for LGPR system described in [24].

$$J = \frac{\sum_{i,d}(A_{i,d}B_{i,d})}{\sqrt{\sum_{i,d}(A_{i,d})^2 \sum_{i,d}(B_{i,d})^2}} \tag{4.1}$$

This registration approach is summarized in Figure 4.2.

Localization, Autonomy, and Seasonal Variation

Later work by Ort et al. built extended LGPR development by Cornick et al. to enable fully autonomous vehicle operation in varying weather conditions using an LGPR sensor [80, 81]. In this development, the same radar array and registration approach from [24] was used. The LGPR measurements were used by the system's localization platform through two instances of an Extended Kalman Filter (EKF). One filter estimates the system velocities from wheel encoder and inertial data and the other filter fuses these velocities with the LGPR-GPS correction.

With the state estimate produced by the two EKFs, a pure pursuit controller is used to drive the vehicle toward the reference path. With this pipeline in place, effective trajectory estimation, planning, and control were enabled by combining proprioceptive information from odometric sensors and exterioceptive information from the LGPR sensor.

One component of this work was to test LGPR performance in varying weather conditions. In the evaluation across different weather conditions, it is apparent that GPR measurements acquired in rain and snow were less correlated to the *a priori* map than in clear weather using the heuristic correlation for registration. It is observed that performance is especially poor when comparing data collected in rainy conditions to the reference map created in clear, likely dry, conditions. These results are summarized in Figures 4.3 and 4.4.

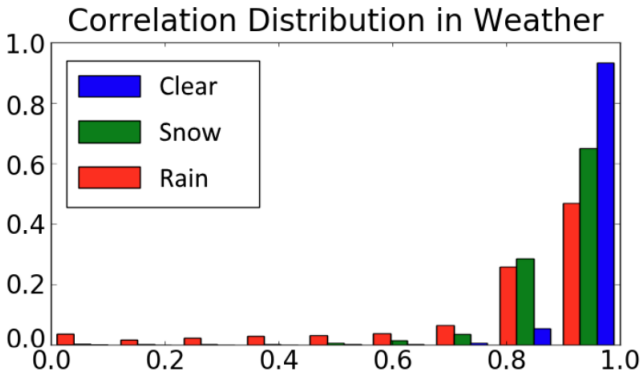

Figure 4.3: Correlation of test run and LGPR map in different weather conditions from [80].

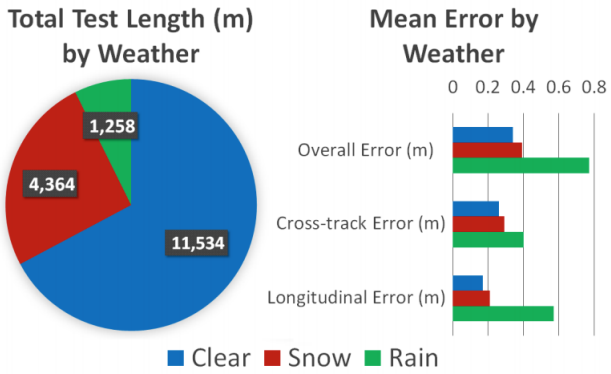

Figure 4.4: Error comparison from ground truth in varying weather conditions from [80].

Engineered Model Baseline

For evaluating our approach, we also consider a baseline *engineered* GPR model that makes use of a correlation metric similar to prior work [24, 80]. This *engineered* GPR model is incorporated as factors in the graph in the same way as our learned GPR model discussed in Section 4.5. In our system, we incorporate the point location of the maximum Pearson product-moment correlation to add as a measurement in the graph. The correlation is defined as:

$$r(A(T), B(T)) = \frac{\sum_{r,c} A_{r,c}(T) B_{r,c}(T)}{\sqrt{\sum_{r,c} A_{r,c}(T)^2} \sqrt{\sum_{r,c} B_{r,c}(T)^2}} \tag{4.2}$$

where, $A_{r,c}(T) = S_{t-k}(T, r, c) - \text{mean}(S_{t-k}(T))$, $B_{r,c}(T) = S_t(T, r, c) - \text{mean}(S_t(T))$, r is the submap row, c is the submap column, and T is the transformation from S_{t-k} to S_t used to only compare the shared region between the submaps.

Specifically, we estimate the transformation along the x-direction of the robot's motion. This is obtained by solving the optimization that maximizes the correlation from Equation 4.2,

$$z_{t-k,t} = K \underset{T}{\text{argmax}} \left(r(A(T), B(T)) \right) \tag{4.3}$$

where, K is a constant that converts submap pixel space to robot motion space.

4.2.2 IMU Preintegration

Our approach uses the method discussed in [39] for integrating inertial measurements and their uncertainty on the manifold $SO(3)$. The IMU receives accelerometer and gyroscope measurements denoted by $z_t^{imu} = [\tilde{a}_t \ \tilde{\omega}_t]$, which are affected by additive white noise η and a time-varying bias b_t.

Preintegrated measurements define relative motion increments that are independent of the pose $s_i = [R_i, \ p_i]$, and velocity v_i of state x_i, which avoids additional computation during relinearization. The preintegrated measurement model is defined as:

$$\begin{aligned} \Delta\tilde{R}_{ij} &= R_i^T R_j \exp(\delta\phi_{ij}) \\ \Delta\tilde{v}_{ij} &= R_i^T (v_j - v_i - g\Delta t_{ij}) + \delta v_{ij} \\ \Delta\tilde{p}_{ij} &= R_i^T \left(p_j - p_i - v_i \Delta t_{ij} - \frac{1}{2} g \Delta t_{ij}^2 \right) + \delta p_{ij} \end{aligned} \tag{4.4}$$

Where the error in the propagated rotation, velocity, and position is: $\eta_{ij}^{\Delta} \doteq [\delta\phi_{ij}, \ \delta v_{ij}, \ \delta p_i j]^T \sim \mathcal{N}(0_{9\times 1}, \Sigma_{ij})$.

$$
\begin{aligned}
\mathbf{r}_{\Delta R_{ij}} &\doteq \log\left(\left(\tilde{R}_{ij}(\bar{b}_i^g)\exp\left(\frac{\partial \Delta\bar{R}_{ij}}{\partial b^g}\delta b^g\right)\right)^T R_i^T R_j\right) \\
\mathbf{r}_{\Delta v_{ij}} &\doteq R_i^T(v_j - v_i - g\delta t_{ij}) \\
&\quad - \left[\Delta\tilde{v}_{ij}(\bar{b}_i^g, \bar{b}_i^a) + \frac{\partial \Delta\bar{v}_{ij}}{\partial b^g} + \frac{\partial \Delta\bar{v}_{ij}}{\partial b^a}\delta b^a]\right] \\
\mathbf{r}_{\Delta p_{ij}} &\doteq R_i^T(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta t_{ij}^2) \\
&\quad \left[\Delta\tilde{p}_{ij}(\bar{b}_i^g, \bar{b}_i^a) + \frac{\partial \Delta\bar{p}_{ij}}{\partial b^g}\delta b^g + \frac{\partial \Delta\bar{p}_{ij}}{\partial b_a}\delta b^a\right]
\end{aligned}
\tag{4.5}
$$

The functions $\Delta\tilde{R}_{ij}(\bar{b}_i^g)$, $\Delta\tilde{v}_{ij}(\bar{b}_i^g, \bar{b}_i^a)$, and $\Delta\tilde{p}_{ij}(\bar{b}_i^g, \bar{b}_i^a)$ are a first-order expansions of Equation 4.4 to relax the constant bias assumption, which are derived in [39].

In order to assume that our noise is sampled from a zero mean Gaussian distribution, we model the slowly time-varying IMU biases as Brownian motion [108]. The change in bias between two consecutive poses is often modeled as:

$$b_j^g = b_i^g + \eta^{bgd}, \quad b_j^a = b_i^a + \eta^{bad} \tag{4.6}$$

where η^{bgd} and η^{bad} have zero mean and covariance $\Sigma^{bgd}\doteq\Delta t_{ij}\mathrm{Cov}(\eta^{bg})$ and $\Sigma^{bad}\doteq\Delta t_{ij}\mathrm{Cov}(\eta^{ba})$. η^{bg} and η^{ba} are Brownian noise terms. The model described in Equation 4.6 can be integrated into the factor graph as:

$$||\mathbf{r}_{b_{ij}}||^2 \doteq ||b_j^g - b_i^g||^2_{\Sigma^{bgd}} + ||b_j^a - b_i^a||^2_{\Sigma_{bad}} \tag{4.7}$$

4.2.3 Learned Sensor Models

Learning-based methods provide an alternate option to model complex sensor measurements. Prior work in visual SLAM has produced dense depth reconstructions from learned feature representations of monocular camera images [17, 26]. In computer vision, spatial correlation networks have been used to learn optical flow and localize RGB cameras in depth maps [19, 20, 38].

Recent work on object state estimation using tactile feedback has demonstrated the effectiveness of learned sensor models in factor graph inference [97]. In this work, Sodhi et al. introduced a learned observation model to predict relative object poses from different keyframes. These predictions are fused with physics and geometric factors in a factor graph framework to estimate latent object poses.

4.3 Learned Feature-based Localization Formulation

4.3.1 Overview of Approach

We propose an approach that allows for correction of accumulated odometry drift without any prior map information. In the absence of a prior map, we must reason over multiple GPR mea-

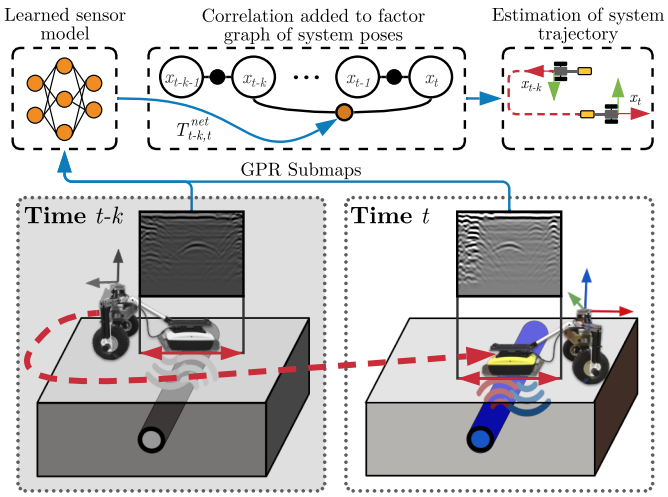

Figure 4.5: Estimating poses for a ground vehicle using subsurface measurements from Ground Penetrating Radar (GPR) as inference over a factor graph. Since GPR measurements are challenging to correlate, a relative transformation is learned between submaps to correct the system's pose.

surements together to be able to infer the latent robot location. We formulate this inference problem using a factor graph, which is now common with many modern localization and SLAM objectives [17, 18, 26, 30, 97]. GPR, inertial, and wheel encoder measurements are incorporated into the graph as factors to estimate the system's latent state, consisting of position, orientation, velocity, and IMU biases. To incorporate measurements into the graph, a sensor model is needed to map measurements to states. For GPR sensors, *a priori* models are typically challenging to obtain since image generation has a complex dependency on subsurface composition and radar physics. Instead, we learn relative sensor models that map non-sequential GPR image pairs to relative robot motion. The relative motion information in turn enables us to correct for drift accumulated when using just proprioceptive sensor information. Our main contributions are:

1. A formulation of the GPR localization problem as inference over a factor graph without a prior map.
2. A learnable GPR sensor model based on submaps.
3. Experimental evaluation on a test platform with a single-channel GPR system operating in three different GPS-denied environments.

Our localization approach is summarized in a supplementary video[1] and in Figure 4.5.

4.3.2 Problem Formulation

We formulate our GPR localization problem as inference over a factor graph. A factor graph is a bipartite graph with two types of nodes: variables $x \in \mathcal{X}$ and factors $\phi(\cdot) : \mathcal{X} \to \mathbb{R}$. Variable nodes are the latent states to be estimated, and factor nodes encode constraints on these variables

[1]Our approach is summarized in this supplementary video `https://youtu.be/HXXgdTJzqyw`

such as measurement likelihood functions.

Maximum a posteriori (MAP) inference over a factor graph involves maximizing the product of all factor graph potentials, i.e.,

$$\hat{x} = \underset{x}{\operatorname{argmax}} \prod_{t=1}^{T} \phi_t(x) \tag{4.8}$$

Under Gaussian noise model assumptions, MAP inference is equivalent to solving a nonlinear least-squares problem [30]. That is, for Gaussian factors $\phi_i(x)$ corrupted by zero-mean, normally distributed noise,

$$\begin{aligned} \phi_t(x) &\propto \exp\left\{-\frac{1}{2}||f_t(x) - z_t||^2_{\Sigma_t}\right\} \\ \Rightarrow \hat{x} &= \underset{x}{\operatorname{argmin}} \sum_{t=1}^{T} ||f_t(x) - z_t||^2_{\Sigma_t} \end{aligned} \tag{4.9}$$

where, $f_t(x)$ is the measurement likelihood function predicting expected measurement given current state, z_t is the actual measurement, and $||\cdot||_{\Sigma_t}$ is the Mahalanobis distance with measurement covariance Σ_t.

For the GPR localization problem, variables in the graph at time step $t = 1 \dots T$ are the 6-DOF robot poses $s_t \in SE(3)$, velocities v_t, and IMU biases b_t, i.e. $x_t = [s_t\ v_t\ b_t]^T$. Factors in the graph incorporate different likelihoods for GPR, IMU, and wheel encoder measurements. At every time step t, new variables and factors are added to the graph. Writing out Eq. 4.9 for the GPR localization objective,

$$\begin{aligned} \hat{x}_{1:T} = \underset{x_{1:T}}{\operatorname{argmin}} \sum_{t=1}^{T} \Big\{ &||f_{gpr}(x_{t\text{-}k}, x_t) - z^{gpr}_{t\text{-}k,t}||^2_{\Sigma_{gpr}} + \\ &||f_{wh}(x_{t\text{-}1}, x_t) - z^{wh}_{t\text{-}1,t}||^2_{\Sigma_{wh}} + \\ &||f_{imu}(x_{t\text{-}1}, x_t) - z^{imu}_{t\text{-}1,t}||^2_{\Sigma_{imu}} \Big\} \end{aligned} \tag{4.10}$$

This objective in Eq. 4.10 is solved online every time step using an efficient, incremental solver iSAM2 [57].

4.4 Submap Construction

Traditional GPR systems provide a one-dimensional trace o_t at each position, which does not provide enough information to effectively determine a system's unique position. A collection of traces represents a local fingerprint that may contain valuable information for localization. Our approach involves constructing submaps S_t based on integration of local wheel encoder measurements w^x_t (Fig. 3.6). A submap is approximated by sampling discrete GPR measurements from a continuous distribution of GPR measurements using an interpolation function $\hat{F}$ to create a uniformly sampled image,

$$S_t = \hat{F}(\{(\bar{o}_t, w^x_t) \mid w^x_{start} \leq w^x_t < w^x_{end}\}) \tag{4.11}$$

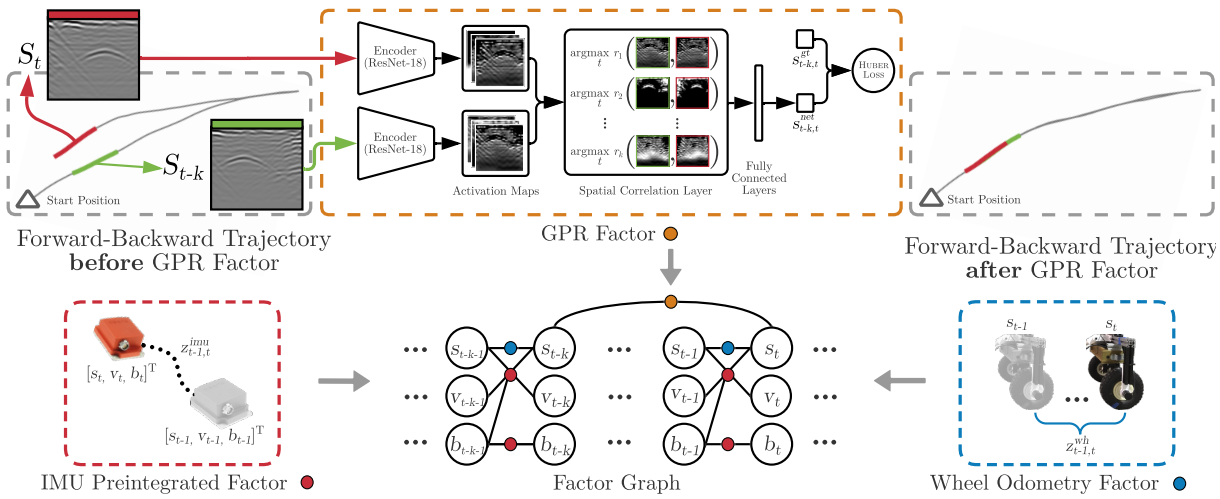

Figure 4.6: Overview of the Ground Encoding system. On the left, two submaps containing similar subsurface structures are acquired during forward-backward trajectory that has accumulated drift. An autoencoder maps these submaps to feature activation maps, which are used by the transform prediction network to find the relative transformation. Factors denoting the measurement likelihoods from GPR, inertial, and wheel encoder measurements are incorporated into the graph to estimate the system's latent states $x_t = [s_t, v_t, b_t]^T \; \forall t \in \{1 \ldots T\}$. The trajectory on the right shows the outcome of the GPR correction.

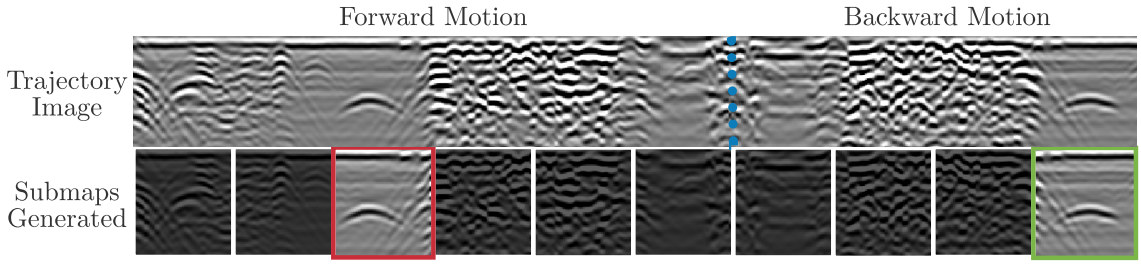

Figure 4.7: Forward-backward trajectory from Figure 4.6 broken down into submaps.

where, $\bar{o}_t$ is the locally averaged GPR measurement, described in Section 3.2, at time t and w_t^x is the wheel encoder measurement at time t.

We use a rule-based method that maps a set of wheel odometry and angular velocity measurements to a decision of whether a short trajectory is a valid submap. This involves ensuring that the z-axis gyroscope reading and y-axis accelerometer measurement are within an acceptable threshold and that wheel encoder measurements are strictly increasing or decreasing, meaning that forward-backward motion is not present in a submap.

In summary, a submap representation is needed to efficiently solve the localization problem since it more adequately models the dependency between neighboring traces in an image. The image-based representation is a subsurface fingerprint that is more descriptive than a one-dimensional trace. Example submaps for a forward-backward trajectory is shown in Figure 4.7.

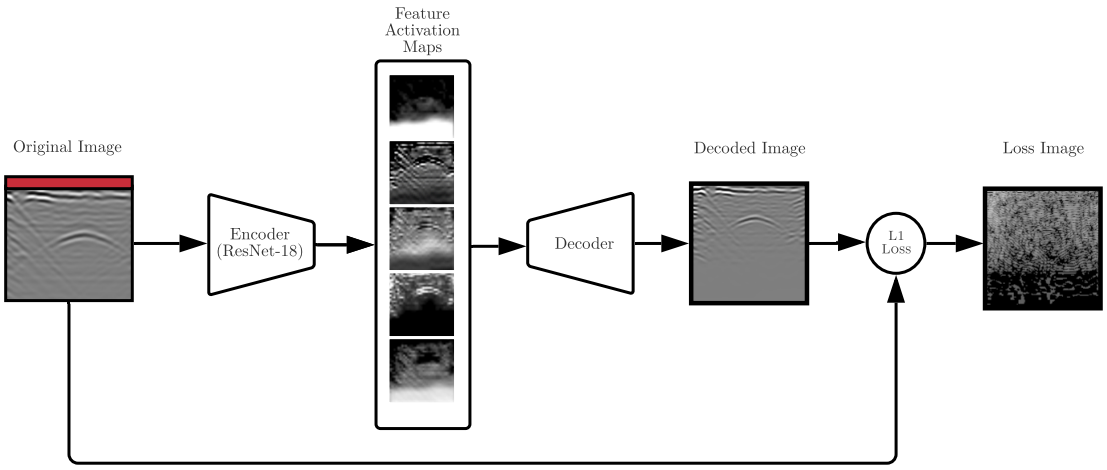

Figure 4.8: Autoencoder architecture used for feature learning. A raw GPR radargram is provided to a ResNet-18 autoencoder, which reduces the image to a series of feature activations. A second ResNet decodes these activation maps to reconstruct the original image.

4.5 Learned Sensor Model

One key insight of our approach is to learn a sensor model directly to register similar GPR submaps. Prior approaches to GPR measurement registration described in [24, 80] rely on particle swarm optimization, which can be non-deterministic, difficult to characterize performance, and computationally expensive [8]. In this approach, we learn a function that maps two non-sequential poses to a relative transformation.

Prior engineered registration approaches perform poorly when:

1. images change when viewed from different angles due to antenna polarization.
2. signal characteristics vary over time.
3. signal characteristics with changes to the physical environment.

Our approach leverages the strengths of the nonparameteric optimization-based methods using a spatial correlation layer, which is used to compare measurements through relative motion in subsampled space. Our pipeline is described in greater detail in the following sections and is summarized in Figure 4.6.

4.5.1 Feature Learning

Identifying the function that directly maps submaps to transformations is prone to overfitting because of noise and aliasing in the original radar submaps. To learn the function that relates two similar submaps S_{t-k} and S_t, an intermediate feature representation f_{t-k} and f_t is needed. A ResNet-18 auto-encoder is used to obtain k feature activation maps that contain relevant features for localization like vertical edges. The autoencoder is trained with an $L1$ reconstruction loss to preserve the sparsity of features in the original submap.

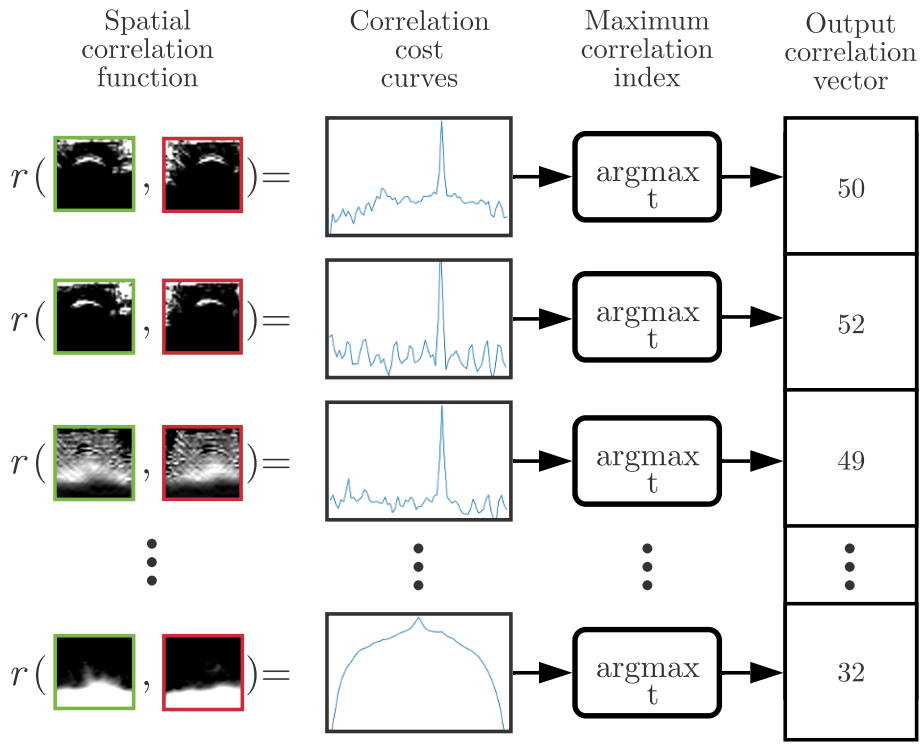

Figure 4.9: Summary of spatial correlation layer for submap transform prediction. Feature activation maps are provided to a spatial correlation function, which computes a curve of image correlations with translation. The maximum values of each curve are compiled into a correlation vector.

4.5.2 Submap Comparison

Prior to identifying transformations in the data, we perform an average pooling on the row-wise standard deviation of submap images to check for salient features. If submaps S_{t-k} and S_t contain valid features, a linear correlation network checks whether they share common features using a method further described in Spatial Correlation and Transform Prediction.

4.5.3 Spatial Correlation

If submaps contain salient features and are correlated, we predict a relative 1D transformation $s^{net}_{t-k,t}$ using a two stage approach. We first construct a set of cost curves by comparing each feature map $S_{t-k,i}$ and $S_{t,i}$ for all k using Equation 4.2 for feature activation map pixel space. We then evaluate the argmax of each cost curve, which is encoded in a vector as shown in Figure 4.9.

Modeling the transformation directly between two images tended to overfit to training data and produced incorrect transform predictions as shown in Figure 4.10(a). The spatial correlation layer simplifies transform prediction by naturally embedding an argmax in the learning process, reducing the learning problem to linear regression as shown in Figure 4.10(b). This approach is generally relevant for predicting transformations between images and volumes for GPR and likely

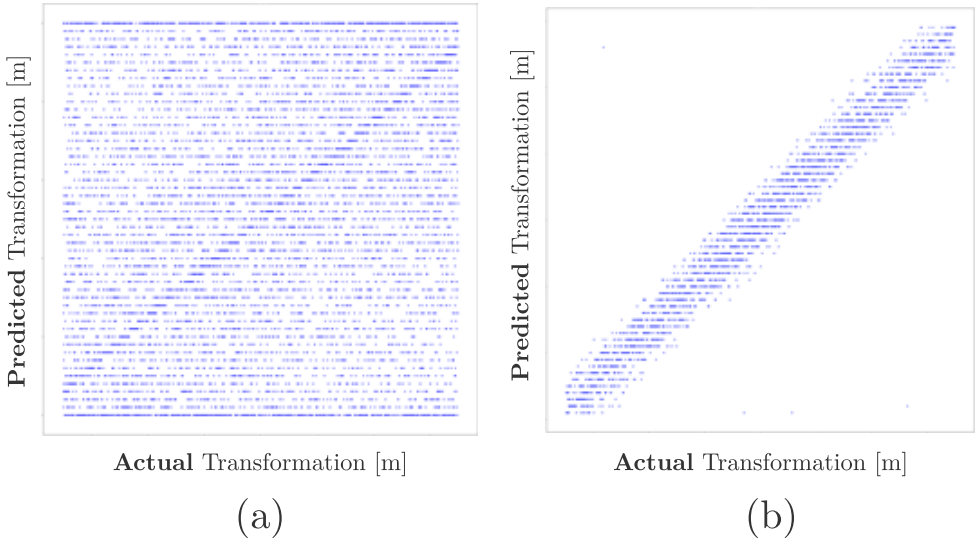

Figure 4.10: Actual versus predicted transformation **(a)** without spatial correlation and **(b)** with spatial correlation. The spatial correlation layer converts it into a linear relationship, simplifying the learning process for the transform prediction network.

other sensing modalities.

While the system produced is staged, alternative approaches can utilize backpropagation through the spatial correlation layer to streamline the learning process [38] and even backpropagation through the graph optimization [113].

4.5.4 Transform Prediction

The transform prediction network learns weights for different filters and a scale factor to convert pixel space to metric range space (e.g. distance in meters). The spatial correlation vector is provided to a fully-connected regression network to identify the relative transformation $s^{net}_{t-k,t}$. Then, the network learns to provide higher weight to filters that are better correlated with the desired translation, reducing the effect of filters that encode common patterns in the data like horizontal banding. The transform prediction network is trained using Huber loss against supervised ground truth data $s^{gt}_{t-k,t}$ acquired by a robotic total station.

4.6 Training the Transform Prediction Network

4.6.1 Ground Truth Positions

The first method identifies pairs of non-sequential poses from a collection of trajectories using ground truth measurements from a robotic total station. Heading estimates were used by computing finite differences between sequential ground truth measurements. Points were placed in a cKD

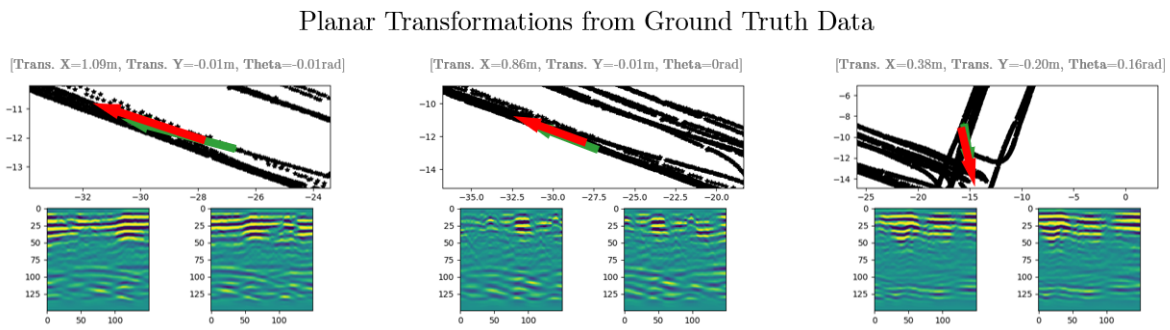

Figure 4.11: Examples of transformations from green coordinate frame to red coordinate frame identified using ground truth data.

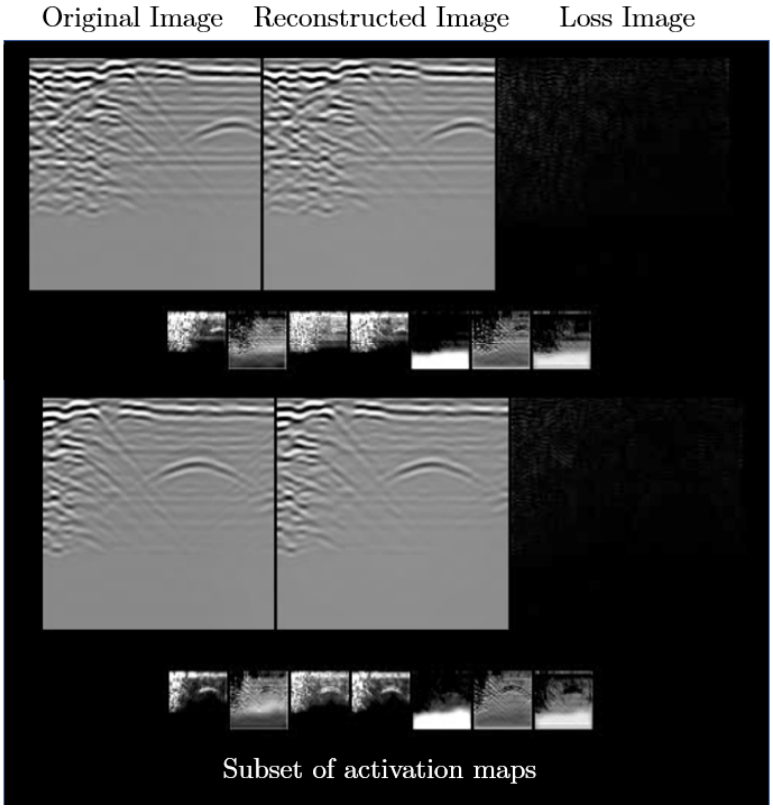

Figure 4.12: Sliding frames from odometry integration depicting synthetic motion. Frames contain the original submap image, reconstruction from the autoencoder, reconstruction loss, and a subset of activation maps.

tree, enabling efficient lookup of neighboring points. Examples of transformations are shown in Figure 4.11.

4.6.2 Artificial Homography

The second method computes artificial homographies between GPR submaps to identify one-dimensional transformations between sequential poses using odometry data. While odometry data

drifts over time, it can provide locally accurate transformation predictions. An example of an artificial homography is shown in Figure 4.12. We found that training using both ground truth and artificial homography methods produced consistent results. However, we decided to use the artificial homography method for training the evaluation transform prediction network since more data can be generated and only one-dimensional transformations were needed.

4.7 Factor Graph Optimization

4.7.1 GPR Factor

Measurements from GPR are incorporated into the graph as factors with the cost from Equation 4.10. The relative GPR factor has a quadratic cost penalizing large residual values and is defined as,

$$||f_{gpr}(x_{t\text{-}k}, x_t) - z_{t\text{-}k,t}||^2_{\Sigma_{gpr}} := ||s_t^{-1} s_{t\text{-}k} \ominus z^{gpr}_{t\text{-}k,t}||^2_{\Sigma_{gpr}} \tag{4.12}$$

where, $z^{gpr}_{t\text{-}k,t}$ is the predicted relative transformation from the transform prediction network, $s_t^{-1} s_{t\text{-}k}$ is the estimated relative pose transformation between two variable nodes in the graph, and $\ominus$ represents the difference between two manifold elements.

4.7.2 IMU Preintegration

For incorporating IMU measurements in the graph, we use the IMU preintegration factor described in Section 4.2.2 and proposed in [39]. The preintegration factor locally integrates multiple high-frequency accelerometer and gyroscope measurements into a single compound preintegrated IMU measurement. This has the advantage of combining the speed and complexity benefits of filtering-based methods along with the accuracy of smoothing methods.

The IMU factor term from Equation 4.10 can be expressed as a residual over the differences in orientation $\Delta R_{i,j}$, velocity $\Delta v_{i,j}$, and position $\Delta t_{i,j}$,

$$||f_{imu}(x_{t\text{-}1}, x_t) - z_{t\text{-}1,t}||^2_{\Sigma_{imu}} = ||\mathbf{r}_{\mathcal{I}_{ij}}||^2 + ||\mathbf{r}_{b_{ij}}||^2 \tag{4.13}$$

where, $\{i, j\}$ are state index pairs between which preintergration is performed. We perform the preintergration between consecutive states, i.e. $\{i, j\} := \{t\text{-}1, t\}$. Here, $\mathbf{r}_{\mathcal{I}_{ij}} = [\mathbf{r}^T_{\Delta R_{i,j}}, \mathbf{r}^T_{\Delta v_{i,j}}, \mathbf{r}^T_{\Delta t_{i,j}}]^T$ is the preintegration error residual and $\mathbf{r}_{b_{ij}}$ is bias term estimation errors. We refer the reader to [39] for more details on these residual error terms.

4.7.3 Wheel Encoder Factor

Wheel encoder measurements are incorporated into the graph between sequential poses $\{s_{t-1}, s_t\}$. The relative wheel encoder factor is defined as,

$$||f_{wh}(x_{t\text{-}1}, x_t) - z^{wh}_{t\text{-}1,t}||^2_{\Sigma_{wh}} := ||s_t^{-1} s_{t\text{-}1} \ominus z^{wh}_{t,t\text{-}1}||^2_{\Sigma_{wh}} \tag{4.14}$$

where, $z^{wh}_{t\text{-}k,t}$ is the relative difference between two poses measured by the wheel encoder.

4.7.4 Optimization

We solve the optimization using the efficient incremental solver iSAM2 [57] at each timestep. Preintegrated IMU measurements are seamlessly added between estimation steps to reduce the size of the estimation problem by better representing high frequency inertial information as described in Section 4.2.2. GPR constraints are added assuming known data association with the measurement obtained from the engineered and learned sensor models.

4.8 Experimental Results

We evaluate our GPR-based localization system in three distinct, GPS-denied environments: a basement (*nsh_b*), a factory floor (*nsh_h*), and a parking garage (*gates_g*). We discussed our experimental setup in Section 3.1. We then provide an ablation of validation losses from different learned and engineered GPR sensor models in Section 4.8.1. We finally evaluate our entire GPR-based localization system and demonstrate the effects of adding GPR information in the graph optimization in Section 4.8.2.

4.8.1 GPR Factor Learning

We now evaluate the performance of different GPR models to be used in the factor graph. Table 4.1 compares losses for the baseline *engineered* models against our learned GPR sensor model discussed in Section 4.5 for different choices of network architectures. The loss here is the mean-squared error that we saw in Equation 5.7 against ground truth transformations. We use the Huber loss function for robustness to outliers. *engineered* is the baseline correlation approach discussed in Section 4.2.1 which resembles prior work [24]. *zeroth* is a zeroth-order model that predicts the average relative transform of the training dataset. *corr-feat* are acquired by concatenating the argmax values of the discrete cost curves from the spatial correlation of autoencoder activation maps as described in Section 4.5. *linear* and *nonlinear* refer to the linear and nonlinear activation functions in the fully-connected layers.

The *engineered* model has notably high losses as there can be many false positive matches since the GPR image data often has similar amplitude with subtle features. In comparison, we see that the different learned GPR model architectures have much lower losses, with *linear, corr-feat* model type having the best performance. We also found that spatial correlation features *corr-feat* generalized much better than using vectorized autoencoder feature maps directly.

4.8.2 Factor Graph Optimization

We finally demonstrate the effect of the GPR sensor model on odometry drift correction through a qualitative and quantitative evaluation. The best performing model from Table 4.1, *linear, corr-feat*, was selected as the *learned* model in this evaluation. We compare the *engineered* and *learned* model with the ground truth trajectory and the results from an *oracle* model. The *oracle*

Table 4.1: Validation losses for different GPR models (cm)

	Dataset			
Model type	gates_g	nsh_b	nsh_h	Combined
engineered	97.66	88.07	100.24	95.33
zeroth	17.81	137.97	23.30	59.49
linear, corr-feat	**14.81**	**2.88**	**3.90**	**7.40**
nonlinear, corr-feat	14.96	3.09	4.06	8.97

model predicts the ground truth relative pose transformation, representing the lower error bound of the GPR factor. The covariance parameters in the graph are fixed and the same across sequences.

4.8.3 Qualitative Evaluation

Both trajectories in Fig. 4.13 were collected in a modern parking garage with homogeneous and repetitious subsurface features, making the test challenging for our GPR system. Additionally, non-deterministic reflections from nearby cars and walls were present in the dataset. As shown, the sensor model relying on *engineered* features identifies incorrect transformations by comparing processed submaps directly, causing substantial drift. This is also reflected in Fig. 4.15, where incorrect loop closures cause the *engineered* model to have greater error than odometry alone. We see that the *learned* model recovers the robot's poses close to the true trajectory and matches the *oracle* performance closely.

4.8.4 Quantitative Evaluation

Fig. 4.14 shows the RMSE of the absolute trajectory error described in [98]. Errors were computed over 7 sequences in *gates_g*, 5 sequences in *nsh_b*, and 3 sequences in *nsh_h* (15 sequences overall). Odometry measurements had greater error in the *gates_g* set since loops and turns were more common. To better compare sensor models, an oracle provided identical loop closure observations to both *engineered* and *learned* models. The *learned* model outperformed the *engineered* model in all sequences. The *engineered* model would often produce false detections in sparsely featured environments, causing a large variance in performance. Repetitive structures and noise in the processed image caused the *engineered* model to perform poorly. The *learned* sensor model performed consistently better by decomposing the image into feature activation maps, containing prominent structures and edges in the data that are easier to compare. The *learned* model nearly performs as well as the *oracle* detector, the model's lower error bound.

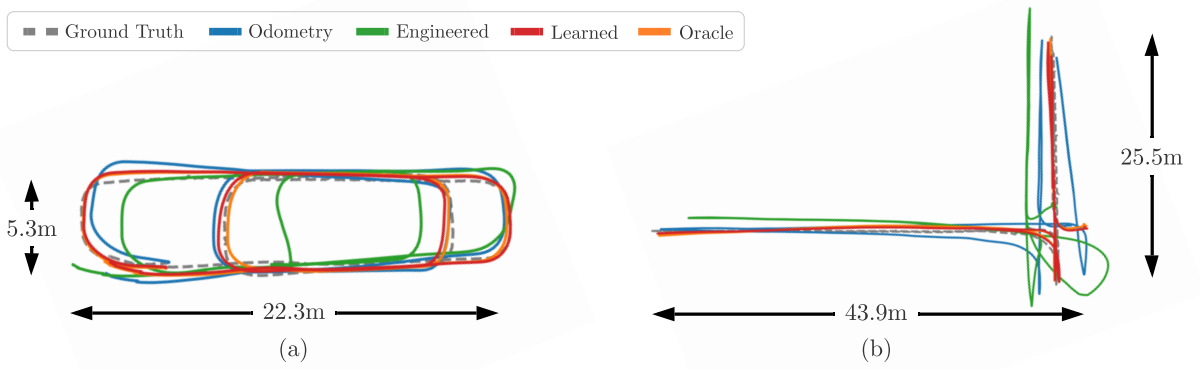

Figure 4.13: Qualitative evaluation of vehicle trajectory estimate over time. **(a)** The robot follows a trajectory that loops over itself twice. **(b)** The robot starts moving on the vertical segment, then moves along the horizontal segment, before finally revisiting the vertical segment. In both cases, the learned sensor model follows the ground truth data better than the engineered method despite receiving the same loop closure detections.

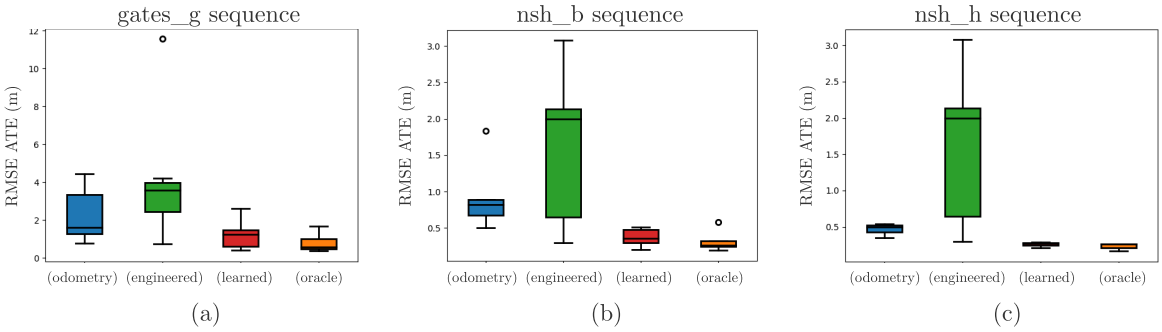

Figure 4.14: RMSE Absolute Trajectory Error (ATE) for each testing environment.

4.8.5 Optimizer Runtime

By leveraging the efficient incremental iSAM2, we were able to obtain real-time performance using localizing GPR. In Figure 4.16, we show 22 minutes of operation consisting of all the sequences in location *nsh_b*. Despite estimating 2,000 latent parameters and the addition of loop closure events, the optimizer runtime does not exceed 0.08 seconds. Spikes in optimizer timing occur at measurement relinearization events.

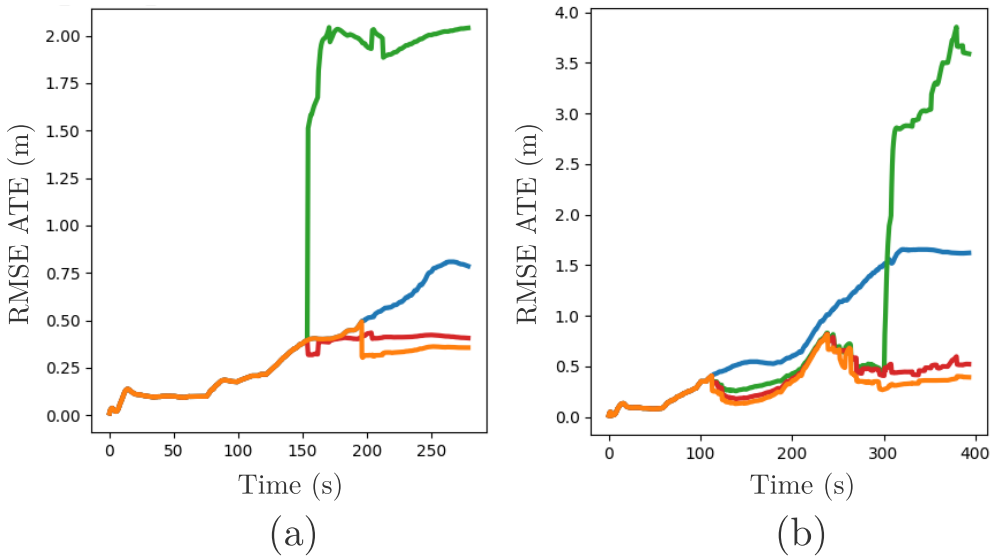

Figure 4.15: RMSE ATE over time. **(a)** Corresponds to the trajectory from Fig. 4.13(a). **(b)** Corresponds to the trajectory from Fig. 4.13(b). In both cases, loop closure events reduce the accumulation of odometry error.

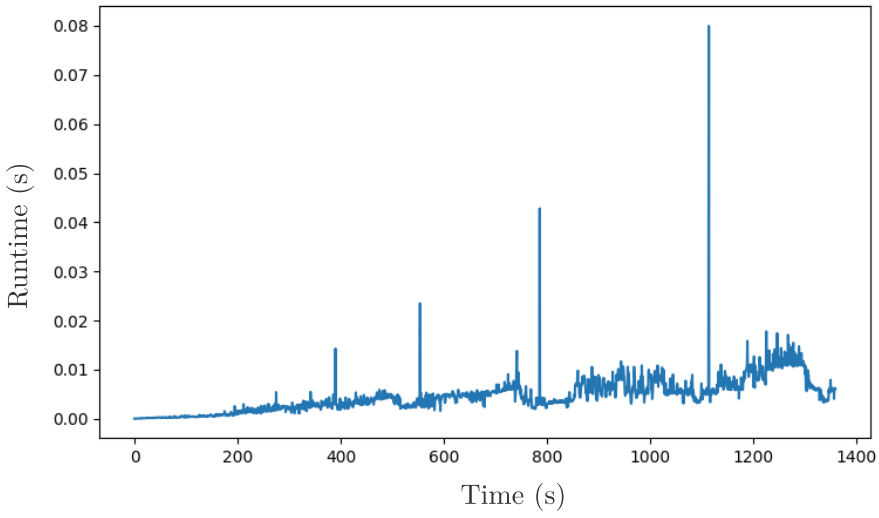

Figure 4.16: Optimizer runtime for entire duration of *nsh_b* trials.

Chapter 5

GPR-based Localization and Mapping

5.1 Introduction

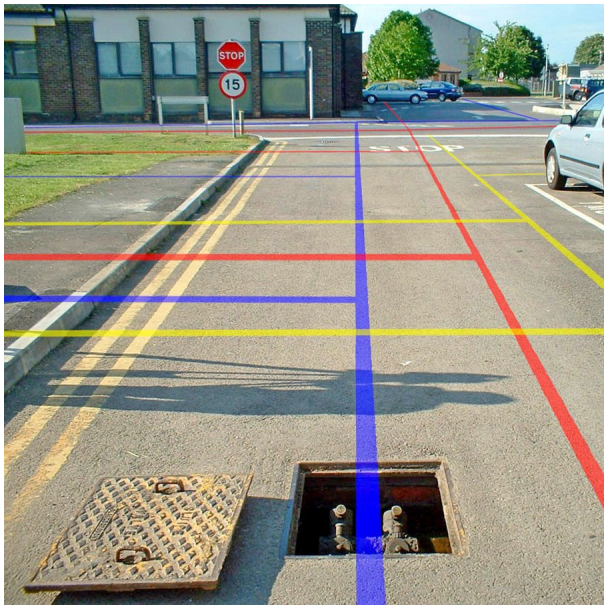

Figure 5.1: Image depicting straight line underground pipes from [82].

In Chapter 4, we propose an approach for GPR localization that requires re-visitation of previous locations in order to correct for accumulated odometric drift. In this chapter, we describe a novel approach for GPR-based positioning that simultaneously estimates robot states and underground landmarks, relaxing the re-visitation constraint. In this approach, we model these landmarks as lines, which are commonly observable as underground pipes [15, 60] and in geology [5]. An example operating environment for this SLAM with lines approach is shown in Figure 5.1. GPR submaps often contain individual observations of these linear features, thus, multiple observations of the same landmark are needed to estimate the landmark line parameters.

In Section 5.2, we describe different line parameters widely adopted by the computer vision and robotics communities. We also describe how lines can be recognized in GPR data. In Section 5.3.2, we describe the structure of the GPR SLAM problem and show how we represent line parameters in a factor graph. In Section 5.4, we describe an observation model for lines in GPR data. In Section 5.5, we derive a custom factor for modeling these line observations in the graph optimization using a Hough line parameterization. In Section 5.6, we show how these line measurements and geometry are incorporated in the graph optimization. Finally, in Section 5.7, we perform a qualitative and quantitative evaluation of the proposed GPR line SLAM method in simulation and real-world settings.

5.2 Background and Related Work

5.2.1 Line Parameters in $\mathbb{R}^2$

Slope intercept form is limited for line parameterizations due to instability when the line slope is infinite, which occurs when a line is vertical. In order to prevent this degenerate case, the normal line parameterization is used. In this form, a line is parameterized by its algebraic distance ρ and angle of its normal θ from the origin. The equation of a line corresponding to this representation is:

$$x_0 \cos\theta + y_0 \sin\theta = \rho \tag{5.1}$$

where x_0 and y_0 is an arbitrary point on the line [33]. This parameterization became popularized in the Hough transform [48] to find lines, curves, and parameterized shapes in images [44, 67, 90].

5.2.2 Line Parameters in $\mathbb{R}^3$

In three-space, Plüker coordinates are an efficient representation for directed lines, which are unique up to a scale factor. This parameterization is summarized in [66]. This representation has been used for visual SLAM with lines [111, 112].

5.2.3 Hyperbola Detection in GPR Data

In Section 2.2.3, we describe that features in GPR B-scans are generically represented as hyperbola. Due to the importance of hyperbola in GPR data, prior work has addressed automatic classification of these features [23, 25, 42, 52, 53, 54, 69, 74, 86].

Many prior approaches use artificial neural networks to identify textures corresponding to hyperbola in images [16, 64, 87]. Some promising work provides additional structure to these networks by adding column-connection clustering (C3), which separates different components of the hyperbola into different clusters [32]. This hyperbola classification system is shown in Figure 5.2. Similar staged systems have also leveraged classical computer vision techniques, like a Histogram of Oriented Gradients feature descriptor, which can be combined with an MLP to

Figure 5.2: Automated hyperbola detection in GPR B-scan from [32]. **(a)** post-processed GPR B-scan. **(b)** Masked image containing hyperbola. **(c)** Hyperbolic curves overlaid on B-scan.

classify hyperbolic features [7]. Other data intensive approaches, such as using a Faster RCNN for classification, have also been used for this application [65].

5.3 GPR Simultaneous Localization and Mapping Formulation

5.3.1 Overview of Approach

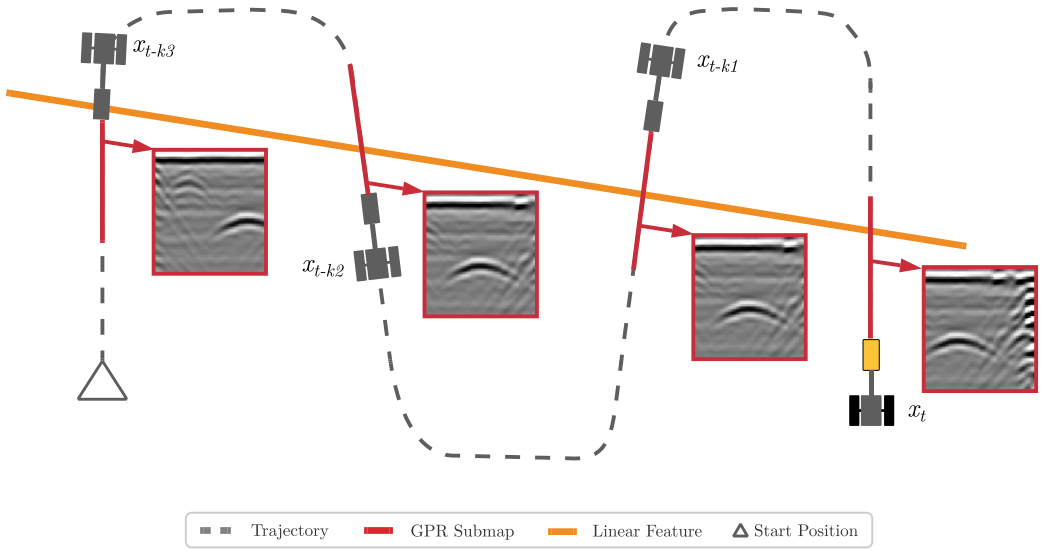

Figure 5.3: Trajectory of robot observing continuous linear feature. The red lines are GPR submaps. In each submap, a prominent hyperbola feature is found, representing an observation of the linear landmark.

Existing approaches to GPR-based positioning require *a priori* models of the robot's operating environment or re-visitation of previously observed spaces. In this approach, we relax this re-visitation constraint by explicitly modeling linear subsurface features as landmarks in a SLAM objective. The system proposed creates submaps as produced in Chapter 4, but is equipped with a line detector, which identifies feature locations in images. Once the feature is localized in the image, the distance from the robot's current position to the linear feature can be computed, which can be added to a custom factor described in Section 5.6 that constrains the robot pose with the line. Our main contributions are:

1. Defining a line feature representation for GPR data using Hough line parameters.
2. Simultaneously infer latent robot states and line parameters in a factor graph optimization.
3. Experimental evaluation in simulated and real-world settings.

A high-level description of our approach is shown in Figure 5.3.

5.3.2 Problem Formulation

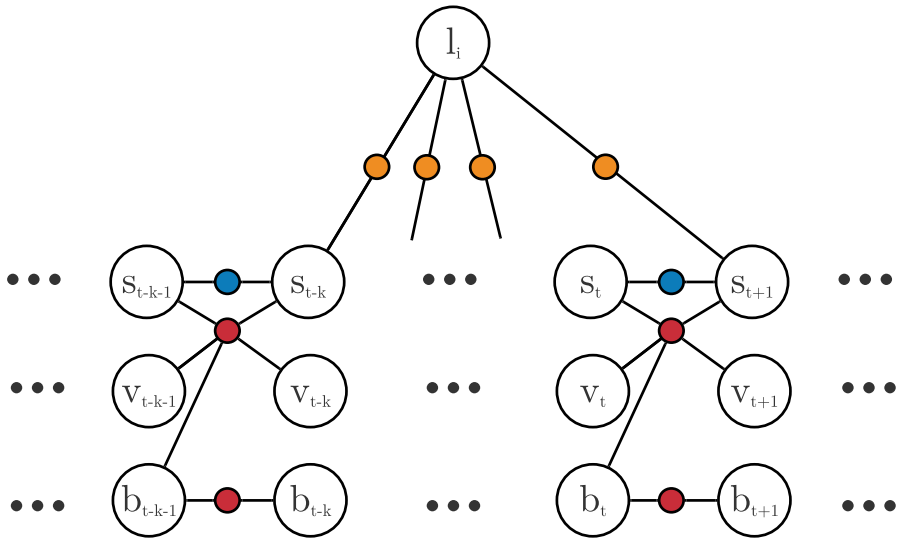

Figure 5.4: Graph of robot states and multiple observations of a single line-feature.

Similar to the problem formulation described in Section 5.3.2, we solve the MAP inference problem by minimizing the following objective:

$$\hat{x}_{1:T},\ \hat{l}_{1:N} = \underset{x_{1:T}}{\operatorname{argmin}} \sum_{t=1}^{T} \Big\{ ||f_{gpr}(l_i, x_t) - z_{i,t}^{gpr}||^2_{\Sigma_{gpr}} + ||f_{wh}(x_{t\text{-}1}, x_t) - z_{t\text{-}1,t}^{wh}||^2_{\Sigma_{wh}} + ||f_{imu}(x_{t\text{-}1}, x_t) - z_{t\text{-}1,t}^{imu}||^2_{\Sigma_{imu}} \Big\} \tag{5.2}$$

where l_i is the i-th landmark observed at time step t. In this instance, $z_{i,t}^{gpr}$ is the distance from the robot's current pose to the line feature identified in the GPR submap. The graph structure depicting this problem is shown in Figure 5.4.

5.4 Line Measurement Model

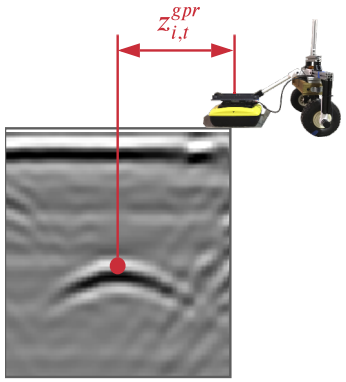

Figure 5.5: Forward distance measurement from robot to feature in GPR submap.

In order to identify the location of lines in GPR submaps, we identify the maximum location of the hyperbola. In experimentation, hyperbola classification is performed manually. However, automated hyperbola extraction algorithms have been extensively explored in literature as described in Section 5.2.3. The distance from the maximum point to the latest measurement is simply the forward distance from the robot to the line. This physical measurement is shown in Figure 5.5.

5.5 Line Geometry

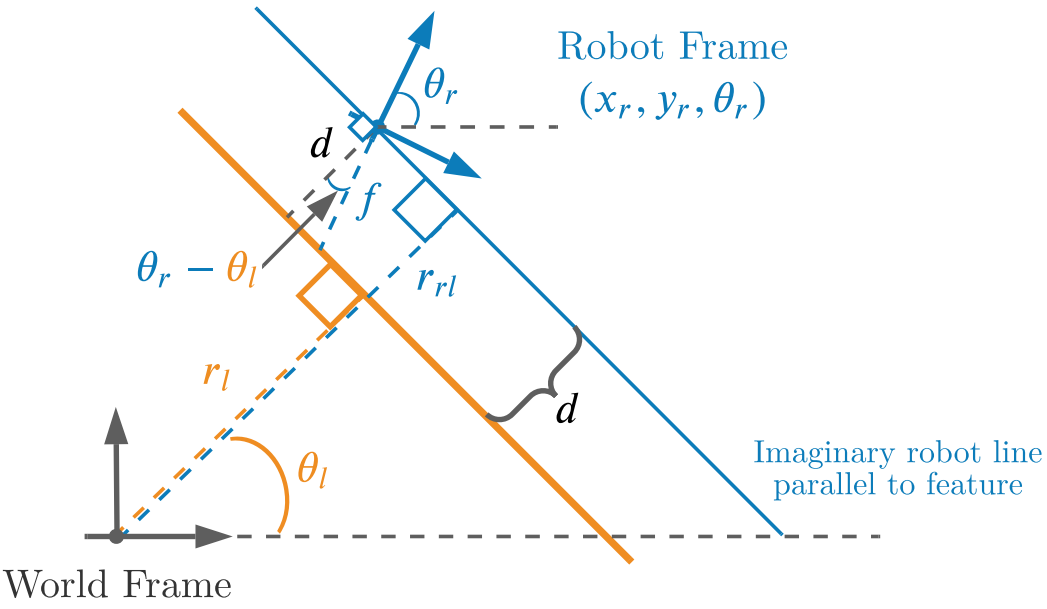

Figure 5.6: Process to compute distance from robot to linear feature as a function of robot pose and line parameters.

We define the line parameters by an angle θ_l and normal length ρ_l as described in Section 5.2.1. The robot is able to observe its forward distance, f, to the line by localizing a linear feature in a submap as described in Section 5.4. We also define an imaginary line r_{rl} that intersects the robot's position (x_r, y_r) in $\mathbb{R}^2$ and share the same angle θ_l. We can then write this imaginary line in normal form, which is shown in Equation 5.3.

$$\rho_{rl} = x_r \cos\theta_l + y_r \sin\theta_l \tag{5.3}$$

Then, we can identify the length along the normal from the feature line to the imaginary robot line as:

$$d = \rho_l - \rho_{rl} \tag{5.4}$$

Given the robot's orientation and the line's orientation, we then know that $\theta_l - \theta_r$ is how much the robot would need to turn to align with the pipe's normal direction θ_l. With this length definition and the robot's pose, we can find the cosine of the difference between the robot's heading and line angle as:

$$\cos(\theta_r - \theta_l) = \frac{d}{f} \tag{5.5}$$

Finally, we solve for f to find the forward distance from the robot to the linear feature to get the desired length from the robot's pose to the ith line feature:

$$f = \frac{d}{\cos(\theta_r - \theta_l)} \tag{5.6}$$

This approach is summarized in Figure 5.6.

5.6 Factor Graph Optimization

Line Factor

Line measurements are incorporated into the factor graph in the first term of Equation 5.2. Given the geometric model described in Section 5.5, we can define the relative GPR-line factor is defined as:

$$||f_{gpr}(l_i, x_t) - z_{i,t}||^2_{\Sigma_{gpr}} \;:=\; ||\frac{\rho_l - (x_r \cos\theta_l + y_r \sin\theta_l)}{\cos(\theta_r - \theta_l)} - z^{gpr}_{i,t}||^2_{\Sigma_{gpr}} \tag{5.7}$$

where $z^{gpr}_{i,t}$ is the forward distance from the robot to the line observed in the submap as described in Section 5.4, $||\cdot||_{\Sigma_{gpr}}$ is the Mahalanobis distance with measurement covariance Σ_{gpr}, and $\ominus$ is the difference between two manifold elements.

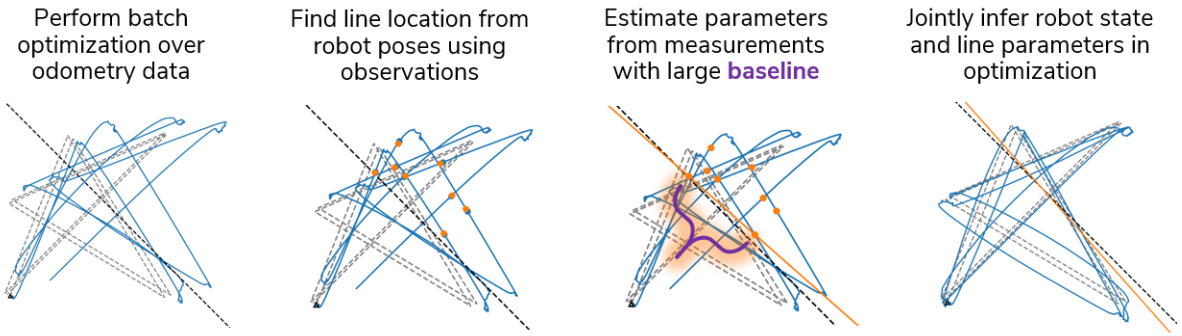

Figure 5.7: Recipe for determining initial values for line parameters in optimization.

Initial Values for Line Parameters

The angular parameter, which is classified as a `Rot2`, is difficult to estimate in the factor graph optimization because of its nonlinear cost landscape. With arbitrary or zero initialization, the angle parameter will often converge to a local minima, which could vary substantially with the true value.

In order to reduce the propensity of this challenge, we introduce a baseline hyperparameter to initialize the length and angle quantities in the optimization. Measurements must be greater than the minimum baseline, a hyperparameter set that describes the minimum acceptable distance between two line observations. Given these two measurements, we can compute an initial estimate of the Hough line parameters, which are provided to the optimizer.

The optimization pipeline is structured in four stages (summarized in Figure 5.7): batch optimization using only odometry data, finding approximate line location given a robot pose and line measurement, selection of candidate line points for initial parameters, and joint optimization with line observaitons and odometry data. The optimization with pure odometry data provides us with approximate locations of line observations. We use Levenberg-Marquardt to perform this optimization after the trajectory sequence as a batch. However, the same process can be applied to the incrementally generated trajectory.

Next, we determine the approximate line location given the estimated robot poses using only odometry data. The submap trajectories are expected to be linear and continuous, meaning that the line position can be obtained using the line measurement described in Section 5.4 and the current robot pose. Therefore the relative transformation from the robot to the line is described by:

$$T_R^L = \begin{bmatrix} I_3 & [z_{i,t}^{gpr}, 0, 0]^T \\ 0_3 & 1 \end{bmatrix} \tag{5.8}$$

Given this definition of the position of the line in the robot's frame, we can obtain the approximate global position of the line by post-multiplying the relative transformation as:

$$T_G^L = s_G^R T_R^L \tag{5.9}$$

The position component of the T_G^L can be obtained by simply extracting the relevant information from the homogeneous transformation. We sequentially compare all other line observations to the first line measurement until the minimum baseline criteria is satisfied. The first points are likely most accurate since the odometry trajectory will drift over time.

Once we have identified our two candidate line observations, we determine the initial point slope form ($y = mx + b$) of the approximate line can be easily obtained. From this equation, we can compute the normal to the line by:

$$\mathbf{n} = [n_y, n_x]^T = \left[-\frac{m}{\sqrt{m^2+1}}, \frac{1}{\sqrt{m^2+1}} \right]^T \tag{5.10}$$

The estimated normal is used to find the approximate Hough angle parameter:

$$\tilde{\theta}_l = \text{atan2}(n_y, n_x) \tag{5.11}$$

The approximate Hough line length along the normal can then be found by:

$$\tilde{r}_l = \frac{b}{\sqrt{m^2+1}} \tag{5.12}$$

These values are finally provided as initial values to the optimization defined in Equation 5.2.

5.7 Experimental Results

5.7.1 Overview of Methods

In experimentation, we compare different approaches: *odometry*, *Localization*, *SLAM without re-visitation (1 line)*, *SLAM without re-visitation (2 lines)*, and *SLAM with re-visitation*. *Odometry* consists of the integration of wheel encoder and IMU measurements. *Localization* uses the method described in Chapter 4. *SLAM without re-visitation (1 line)* estimates the line-feature parameters along with the robot state in the optimization as described in Section 5.3.2 with one line. *SLAM without re-visitation (2 line)* estimates the line-feature parameters along with the robot state in the optimization as described in Section 5.3.2 with two lines. *SLAM with re-visitation* jointly uses the constraints from *localization* and the line observations *SLAM without re-visitation (1 line)* in the optimization.

5.7.2 Simulated Experiments

Simulation Environment

To have accurate ground-truth data about the location of linear features, we created a planar simulation environment (operating in $SE(2)$). In this environment, a robot follows a trajectory described by a series of waypoints and outputs simulated odometry and distance-to-line measurements.

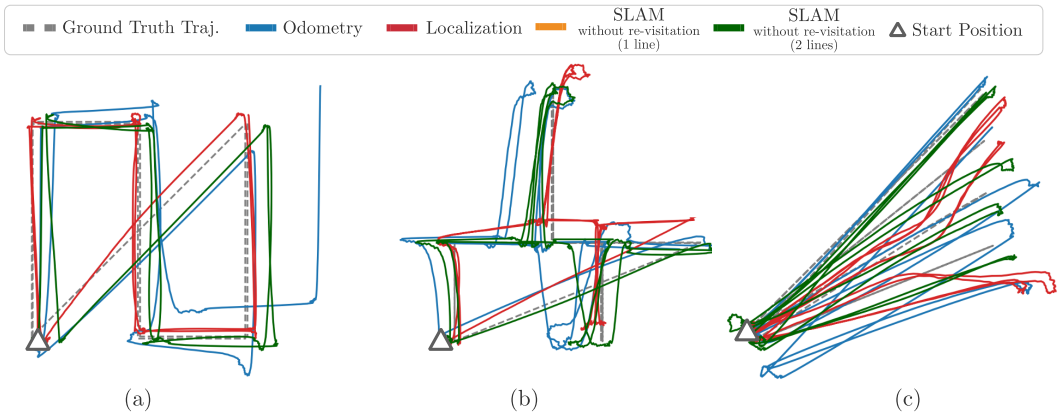

Figure 5.8: Qualitative comparison of simulated trajectories. Localization and SLAM approaches in **(a)** and **(b)** improve the trajectory estimate. *SLAM without re-visitation (1 line)* is not shown for figure clarity.

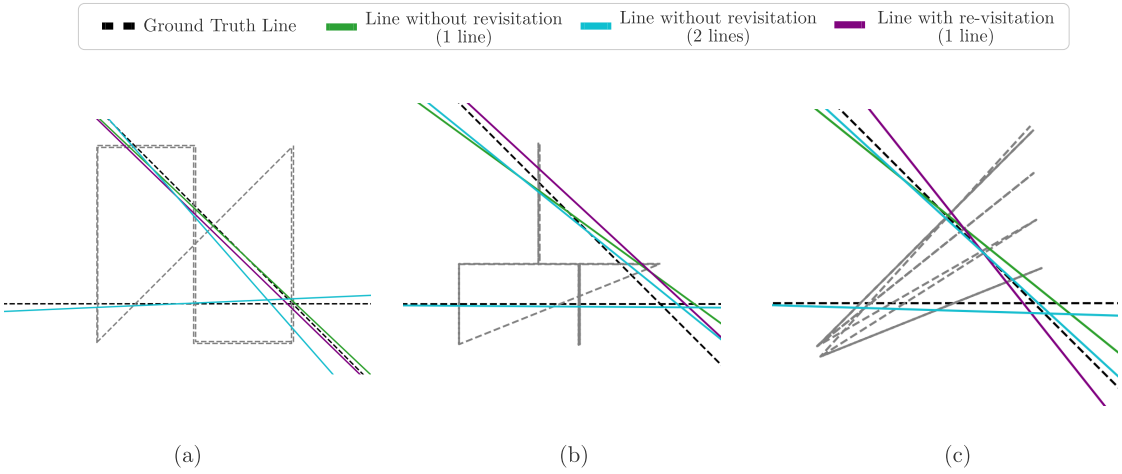

Figure 5.9: Qualitative comparison of estimated line features from trajectories in Figure 5.8. In **(a)** and **(b)**, the line parameters are well-estimated. The line with re-visitation approach yields improved performance since the angle parameter is initialized closer to the true value.

Qualitative

The *SLAM without re-visitation (2 lines)* approach typically tracks the ground truth trajectory best. The *SLAM with re-visitation* trajectory often performed slightly better than the *localization* approach for cases where re-visitation occurs. In all cases, localization and SLAM approaches performed better than *odometry* alone as expected. One particular reason why *SLAM without re-visitation (1 line)* does not perform as well as *localization* is because the factor only reduces error normal to the line feature. In the presence of only one line, drift will still accumulate along the other axis. This problem is partially corrected in *SLAM without re-visitation (2 lines)*.

Additionally, the *SLAM with re-visitation* method produces better line estimates since the initial trajectory estimate is more accurate than the *odometry* trajectory. Since the initial trajectory

estimate is more accurate, the line is initialized with more correct initial Hough parameters. See Section 5.6 for additional context on line initialization.

SLAM without re-visitation does not perform well in cases with long trajectories with a small baseline exists, since drift will continuously acrue along the line.

An ablation of three estimated trajectories with these approaches is shown in Figure 5.8. The corresponding line-features estimated for *SLAM without re-visitation* and *SLAM with re-visitation* are shown in Figure 5.9.

Quantitative

For these different approaches, we evaluated absolute trajectory error (ATE) similar to experimentation performed in Section 4.8. The *SLAM without re-visitation (2 lines)* approach typically had the lowest ATE. The ATE for *localization* and *SLAM with re-visitation* performed better then *SLAM without re-visitation (1 line)* since more re-visitation events occurred over line observations. The *SLAM without re-visitation (1 line)* approach reduced error compared to odometry, which is particularly meaningful when the trajectory does not contain re-visitation events. These results are compared in Figure 5.10.

In addition to comparing ATE, we additionally compare the error of the Hough line feature parameter estimates. The *SLAM with re-visitation* produces more accurate line length estimates compared with *SLAM without re-visitation (1 line)*. In *SLAM with re-visitation*, the weight applied to the *localization* component is higher than the weight applied to line observations. This causes the robot to rely less on the line parameters, which converge to the *localization* trajectory. *SLAM without re-visitation (2 lines)* has higher line length error since there are more degrees of freedom in the optimization, meaning that there can be multiple line lengths that reduce error.

The results for the line angle parameter differ from that of the line length parameter. The *SLAM with re-visitation* approach performs better than *SLAM without re-visitation (1 line)* approach. This occurs since the first batch optimization used to determine the initial line parameters, as described in Section 5.6, is more accurate. This is because the *localization* typically is more accurate than the *odometry* trajectory, leading to better initial values for the line parameters. *SLAM without re-visitation (2 lines)* performs better than *SLAM without re-visitation (1 line)*, but not as well as *SLAM with re-visitation*. Thus, we observe that the performance of line angle parameter estimation is coupled with good initial value estimation.

5.7.3 Real-World Results

Test Setting

The real-world test described in this section was performed in a flat parking garage environment (*gates_g* location). The data is available as a part of the CMU-GPR dataset described in Section 3.3. We used the SuperVision experimental test rig for data acquisition and a robotic total station for

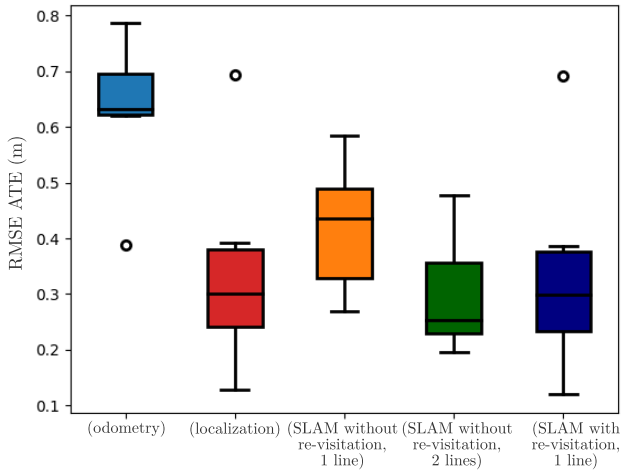

Figure 5.10: RMSE Absolute Trajectory Error (ATE) for different GPR-based positioning model with simulated data.

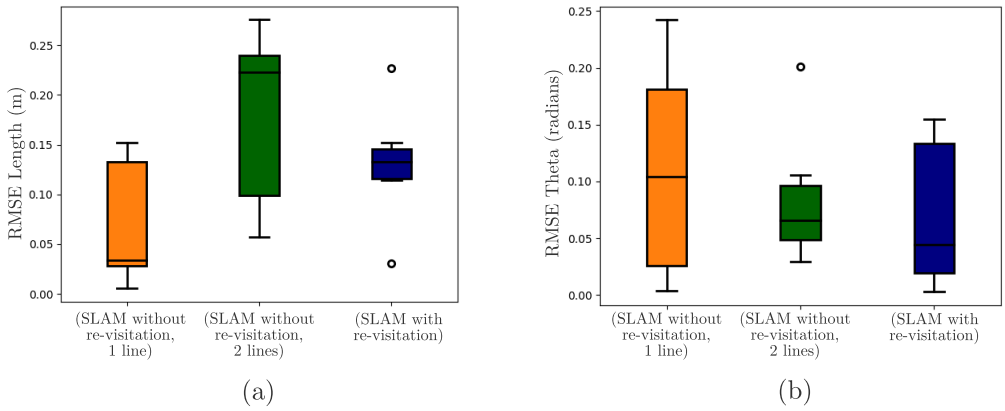

Figure 5.11: RMSE of line of estimated Hough line parameters for simulated data.

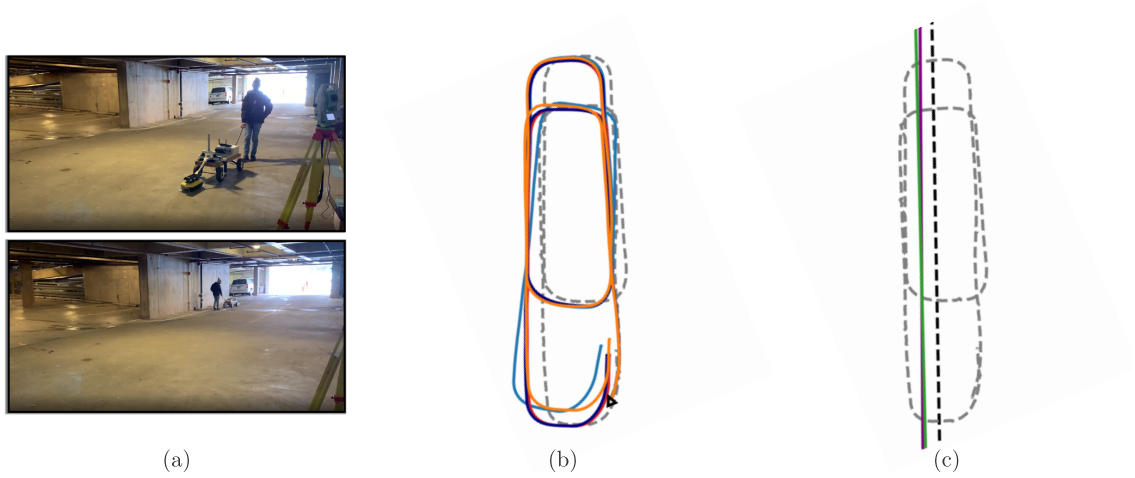

Figure 5.12: Qualitative comparison of SLAM approaches for GPR-based positioning in real-world environment. **(a)** two images of the test environment. **(b)** estimated trajectories using GPR-based SLAM methods. **(c)** estimated features from GPR-based SLAM methods.

ground truth position information as described in Section 3.1. Images of the test environment are shown in Figure 5.12(a). The ground truth line chosen is an approximation of a feature observed in the experimental GPR data.

Qualitative Results

When comparing trajectory types, it is apparent that the trajectory estimated with *SLAM without re-visitation* is more consistent with the ground truth trajectory perpendicular to the line feature. *Localization* and *SLAM with re-visitation* produced similar results and better aligned with the entire trajectory. This is expected since the trajectory contains re-visitation events. This approach reduces error over both dimensions as opposed to a single dimension from modeling line observations. The qualitative trajectory results of the different SLAM approaches are shown in Figure 5.12(b).

Both *SLAM without re-visitation* and *SLAM with re-visitation* approximate the Hough angle parameter well, but have large error in the normal line length. While error does exist, the approximated line appears to intersect with the estimated trajectory approximately where the true line intersects the ground truth trajectory. Drift orthogonal to the line causes the estimated trajectory to appear oversized, which is not corrected by line observations. The qualitative line feature results of the different SLAM approaches are shown in Figure 5.12(c).

Quantitative Results

ATE and RMSE for Hough line parameters are shown for this example in Table 5.1. Since the trajectory contained several loops, the *localization* approach performed best as expected. SLAM without re-visitation produced results better than odometry, but not as well as *SLAM with re-visitation* since error is only constrained the dimension orthogonal to the line. Both

Table 5.1: Trajectory and landmark error for real-world testing

Traj. Type	RMSE ATE [m]	RMSE Length [m]	RMSE Theta [rad]
Odometry	0.803	-	-
Localization	0.464	-	-
SLAM without re-visitation	0.618	1.045	0.005
SLAM with re-visitation	0.487	0.787	0.012

SLAM approaches produced similar estimates for the Hough angle parameter, but did not well approximate the normal length as described previously.

Chapter 6

Conclusion

6.1 Contributions

In this thesis, we address the challenge of using underground information for robot localization and mapping. We propose novel approaches for robot localization and feature-based SLAM using GPR sensors and demonstrate improved positioning performance by incorporating subsurface information with odometric sensors. These algorithms were tested using a custom experimental rig outfitted with a low-cost, off-the-shelf, ground-coupled, single-channel GPR sensor. While our evaluation was performed with a ground-coupled GPR system, our work can generalize for air-launched GPR systems, improving the viability for real-world deployment. These methods can also be applied to multi-channel systems, which generate 3D models of the underground environment as opposed to 2D images. Some examples of these types of systems are shown in Figure 6.1.

In Chapter 3, we describe the design, development, and testing of our experimental rig and signal processing software. We detail some challenges of GPR antenna placement, the differences between global and local information in GPR data, and signal pre-processing methods to improve data quality for GPR-based localization and mapping. We also describe our open source CMU-GPR dataset, which is available to researchers interested in GPR-based localization and mapping.

Figure 6.1: Different air-coupled and multi-channel GPR systems from [1, 2, 80] respectively.

In Chapter 4, we present a GPR-based localization system that uses a learned sensor model to position a robot in unknown, GPS-denied environments. We created submaps describing a robot's local environment by horizontally stacking GPR measurements. We then used an autoencoder to find low-level feature embeddings that describe specific features in GPR submaps. We identified that spatial correlation reduces the feature activations to a correlation vector, simplifying the learning problem to linear regression. The correlation vector is provided to a transform prediction network to predict relative pose transformations between non-sequential submaps. These transformations were then incorporated as measurement likelihoods in a factor graph, where the GPR measurements corrected for accumulated drift of proprioceptive IMU and wheel encoder measurements. We validated our system in three distinct environments, where we showed improved localization performance over existing correlation-based approaches.

In Chapter 5, we proposed a SLAM approach that models GPR observations of linear underground structures, like pipes. In this approach, we jointly infer robot states and Hough line parameters in the factor graph optimization. We show that by modeling individual line observations, we can correct for odometric drift perpendicular to line features and enable GPR-based positioning without re-visitation. Additionally, we describe a method to identify initial values for line parameters to improve the likelihood of the optimizer convergence to the true line parameters. We compare our GPR line SLAM approach with the prior localization-based approach in simulated and real-world environments in sequences with re-visitation events.

6.2 Observations and Future Work

We see many avenues for extending the use of GPR in robotics and in challenging, unstructured, real-world settings. In this work, we implement methods to reason about underground information, which can improve the robustness of GPR-based positioning systems.

In Chapter 4, we represent GPR data as 2D image submaps. We observe that a 2D image representation performs well compared to raw 1D traces, which do not provide unique enough features, and 3D point clouds, which are sensitive to user-defined parameters, noise, and variable subsurface composition. Additional work can leverage structures, such as points, curves, and hyperbolae in GPR data to find improved representations for GPR data. Our learned sensor model is trained using data collected at a single point in time and has not been tested for long-term prior map registration. A similar approach can be used for long-term GPR scan registration by identifying how features are affected by changes in subsurface moisture content. In the future, we would like to extend this work by using data collected from highly accurate, multi-channel GPR systems during different seasons and weather conditions to improve the robustness of prior map registration. This is now feasible with the release of the GROUNDED dataset [81]. An example of data from GROUNDED is shown in Figure 6.2.

In Chapter 5, we represent GPR data by line observations. In this work, we assume that features are linear, which can be locally correct, but is not globally realistic. Future work should model arbitrary curves with finite extent in the estimation problem. Estimation of parameters

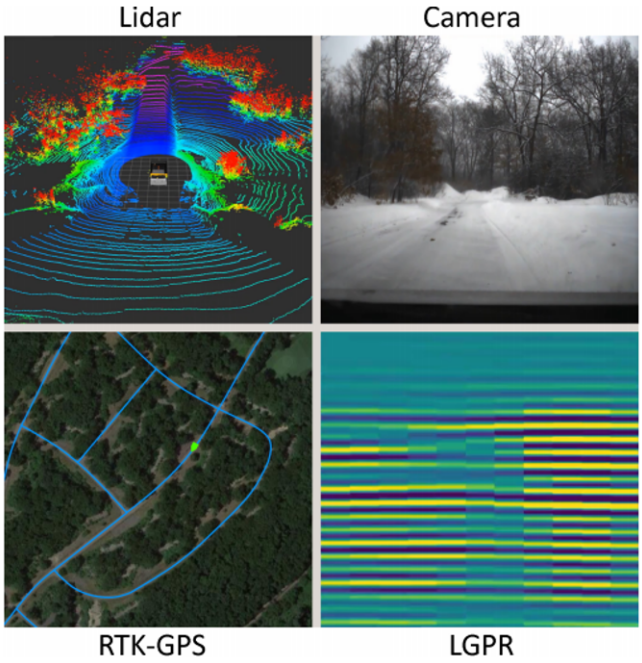

Figure 6.2: Data from GROUNDED localizing GPR dataset. Retrieved from [81].

has been explored in plane estimation [56, 110], lines [112], and curves in visual SLAM [73]. However, one variation that GPR-based positioning approaches must consider is that these parameters can often only be estimated by acquiring multiple measurements. Additionally, future work can model line observations in three-space and leverage line parameterizations like Plüker coordinates [66]. One limitation of the current Hough line parameterization is the challenge of scale in the optimization. Specifically, when the line length is large, small changes in angle cause large changes in line properties. This can be partially resolved using local line parameterizations, where the line can be defined in the local coordinate frame of its first robot observation.

An additional dimension to the GPR-based perception problem includes navigation in semi-static environments. While some work has addressed long-term navigation with spatio-temporal change [34, 68, 89], additional work can address how to model discrete and continuous changes of environments over time. An example of this in the context of GPR is modeling the addition of new underground features, like pipes, when they are observed.

GPR is a tool that can help enable robust navigation in unknown and challenging environments, where traditional sensors typically fail. Future work should address more general representations for underground information to help enable the adoption and deployment of real-world GPR-based perception systems.

Bibliography

[1] 3D-Radar. http://3d-radar.com/. (document), 6.1

[2] Ground penetrating radar. http://www.erikuab.com/ground-penetrating-radar/. (document), 6.1

[3] Contributors. In Harry M. Jol, editor, *Ground Penetrating Radar Theory and Applications*, pages xv – xviii. Elsevier, Amsterdam, 2009. ISBN 978-0-444-53348-7. doi: https://doi.org/10.1016/B978-0-444-53348-7.00018-1. URL http://www.sciencedirect.com/science/article/pii/B9780444533487000181. 2.2.1, 2.2.1, 2.2.3

[4] Roberto Aldera, Daniele De Martini, Matthew Gadd, and Paul Newman. What Could Go Wrong? Introspective Radar Odometry in Challenging Environments. In *IEEE Intelligent Transportation Systems (ITSC) Conference*, Auckland, New Zealand, October 2019. 2.3

[5] R. Anand Krishnan, S. Anbazhagan, S. Anirudhan, Jamal Ansari, M. Arindam, K. Brindha, J.Y. Chen, S. Chidambaram, M. Chinnamuthu, Sang Yong Chung, V. Dhanu Radha, L. Elango, Hussam Eldin Elzain, C. Gajendran, N. Ganesh, G. Gnanachandrasamy, S. Gopinath, Cüneyt Güler, K. Jesuraja, Madan Kumar Jha, Cincy John, A. Jothibasu, D. Karunanidhi, Haris Hasan Khan, Arina Khan, Rahman Khatibi, Deepesh Machiwal, N.S. Magesh, S. Manikandan, A. Manisha, N. Murali Krishnan, C.K. Muthumaniraja, Keyvan Naderi, Ata Allah Nadiri, Banajarani Panda, C.R. Paramasivam, R. Prakash, M.V. Prasanna, M. Rajendran, R. Rajesh, A.L. Ramanathan, T. Ramkumar, R.G. Rejith, P.M. Saharuba, K. Saravanan, Michael Schneider, S. Selvam, S. Venkatramanan, Priyanka Sharma, Prafull Singh, K. Srinivasamoorthy, M. Sundararajan, P. Thamarai, R. Thilagavathi, C. Thivya, Frank T.-C. Tsai, Fatemeh Vahedi, S. Vasudevan, Pradipika Verma, and J. Vidhya. Contributors. In Senapathi Venkatramanan, Mohan Viswanathan Prasanna, and Sang Yong Chung, editors, *GIS and Geostatistical Techniques for Groundwater Science*, pages xiii–xiv. Elsevier, 2019. ISBN 978-0-12-815413-7. doi: https://doi.org/10.1016/B978-0-12-815413-7.09992-2. URL https://www.sciencedirect.com/science/article/pii/B9780128154137099922. 5.1

[6] A. P. Annan and S. Cosway. Ground penetrating radar survey design. 1999. (document), 3.5

[7] Pouria Asadi, Mayrai Gindy, Marco Alvarez, and Alireza Asadi. A computer vision based rebar detection chain for automatic processing of concrete bridge deck gpr data. *Automation in Construction*, 112:103106, 2020. ISSN 0926-5805. doi: https://doi.org/10.1016/j.autcon.2020.103106. URL https://www.sciencedirect.com/science/article/pii/S0926580519306259. 5.2.3

[8] Drew Bagnell, Greg Seyfarth, and Zachary Batts. Good, bad, and ugly of particle filters, 2014. 4.5

[9] Alexander Baikovitz, Paloma Sodhi, Michael Dille, and Michael Kaess. CMU-GPR Dataset: Ground penetrating radar dataset for robot localization and mapping, 2021. 3.3

[10] Dan Barnes and Ingmar Posner. Under the radar: Learning to predict robust keypoints for odometry estimation and metric localisation in radar. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Paris, 2020. URL `https://arxiv.org/abs/2001.10789`. 2.3

[11] Dan Barnes, Rob Weston, and Ingmar Posner. Masking by moving: Learning distraction-free radar odometry from pose information. In *Conference on Robot Learning (CoRL)*, 2019. URL `https://arxiv.org/pdf/1909.03752`. 2.3

[12] Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, Paris, 2020. 2.3

[13] I. Bekkerman and J. Tabrikian. Target detection and localization using mimo radars and sonars. *IEEE Transactions on Signal Processing*, 54(10):3873–3883, 2006. doi: 10.1109/TSP.2006.879267. 2.3

[14] A. Benedetto and L. Pajewski. *Civil Engineering Applications of Ground Penetrating Radar*. Springer Transactions in Civil and Environmental Engineering. Springer International Publishing, 2015. ISBN 9783319048130. URL `https://books.google.com/books?id=jtX5BwAAQBAJ`. 2.2.1

[15] Anita Bernatek-Jakiel and Marta Kondracka. Detection of soil pipes using ground penetrating radar. *Remote Sensing*, 11:1864, 08 2019. doi: 10.3390/rs11161864. 5.1

[16] Sven Birkenfeld. Automatic detection of reflexion hyperbolas in gpr data with neural networks. In *2010 World Automation Congress*, pages 1–6, 2010. 5.2.3

[17] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J. Davison. CodeSLAM - learning a compact, optimisable representation for dense visual SLAM. *CoRR*, abs/1804.00874, 2018. URL `http://arxiv.org/abs/1804.00874`. 4.2.3, 4.3.1

[18] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016. doi: 10.1109/TRO.2016.2624754. 2.1.1, 2.1.2, 4.3.1

[19] D. Cattaneo, M. Vaghi, A. L. Ballardini, S. Fontana, D. G. Sorrenti, and W. Burgard. CMRNet: Camera to LiDAR-map registration. *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Oct 2019. doi: 10.1109/itsc.2019.8917470. URL `http://dx.doi.org/10.1109/ITSC.2019.8917470`. 4.2.3

[20] Daniele Cattaneo, Domenico Giorgio Sorrenti, and Abhinav Valada. CMRNet++: Map and camera agnostic monocular visual localization in LiDAR maps, 2020. 4.2.3

[21] Sarah Cen and Paul Newman. Radar-only ego-motion estimation in difficult settings via

graph matching. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Montreal, Canada*, 2019. 2.3

[22] Sarah H. Cen and Paul Newman. Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions. *Proceedings of the 2018 IEEE International Conference on Robotics and Automation*, 2018. 2.3

[23] G Ciochetto, S Delbo, P Gamba, and D Roccato. Fuzzy shell clustering and pipe detection in ground penetrating radar data. In *IEEE 1999 International Geoscience and Remote Sensing Symposium. IGARSS'99 (Cat. No. 99CH36293)*, volume 5, pages 2575–2577. IEEE, 1999. 5.2.3

[24] Matthew Cornick, Jeffrey Koechling, Byron Stanley, and Beijia Zhang. Localizing Ground Penetrating RADAR: A Step Toward Robust Autonomous Ground Vehicle Localization. *Journal of Field Robotics*, 33(1):82–102, jan 2016. ISSN 15564959. doi: 10.1002/rob.21605. (document), 4.1, 4.2.1, 4.2.1, 4.2, 4.2.1, 4.2.1, 4.5, 4.8.1

[25] E Costamagna, P Gamba, and S Lossani. A neural network approach to the interpretation of ground penetrating radar data. In *IGARSS'98. Sensing and Managing the Environment. 1998 IEEE International Geoscience and Remote Sensing. Symposium Proceedings.(Cat. No. 98CH36174)*, volume 1, pages 412–414. IEEE, 1998. 5.2.3

[26] Jan Czarnowski, Tristan Laidlow, Ronald Clark, and Andrew J Davison. DeepFactors: Real-time probabilistic dense monocular SLAM. *IEEE Robotics and Automation Letters*, 5 (2):721–728, 2020. 4.2.3, 4.3.1

[27] David J. Daniels, editor. *Ground Penetrating Radar*. Radar, Sonar amp; Navigation. Institution of Engineering and Technology, 2004. URL `https://digital-library.theiet.org/content/books/ra/pbra015e`. 2.2.1

[28] Jeffrey Daniels. Ground penetrating radar fundamentals. 01 2000. doi: 10.4133/1.2921864. 2.2.1

[29] Alex Davies. The clever way Ford's self-driving cars navigate in snow. (document), 1.2

[30] Frank Dellaert and Michael Kaess. Factor graphs for robot perception. *Foundations and Trends in Robotics*, 6(1-2):1–139, 2017. (document), 2.2, 2.1.3, 4.3.1, 4.3.2

[31] Nectaria Diamanti and Peter Annan. Air-launched and ground-coupled gpr data. 03 2017. doi: 10.23919/EuCAP.2017.7928409. 2.2.3

[32] Qingxu Dou, Lijun Wei, Derek R. Magee, and Anthony G. Cohn. Real-time hyperbola recognition and fitting in gpr data. *IEEE Transactions on Geoscience and Remote Sensing*, 55(1):51–62, 2017. doi: 10.1109/TGRS.2016.2592679. (document), 5.2.3, 5.2

[33] Richard O. Duda and Peter E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, January 1972. ISSN 0001-0782. doi: 10.1145/361237.361242. URL `https://doi.org/10.1145/361237.361242`. 5.2.1

[34] Marcin Dymczyk, Elena Stumm, Juan Nieto, Roland Siegwart, and Igor Gilitschenski. Will it last? learning stable features for long-term visual localization. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 572–581, 2016. doi: 10.1109/3DV.2016.66. 6.2

[35] Maurizio Ercoli, Cristina Pauselli, Alessandro Frigeri, Emanuele Forte, and Costanzo Federico. 3-D GPR data analysis for high-resolution imaging of shallow subsurface faults: the Mt Vettore case study (Central Apennines, Italy). *Geophysical Journal International*, 198(1):609–621, 05 2014. ISSN 0956-540X. doi: 10.1093/gji/ggu156. URL https://doi.org/10.1093/gji/ggu156. 1

[36] X. Feng, M. Sato, Y. Zhang, C. Liu, F. Shi, and Y. Zhao. CMP antenna array GPR and signal-to-clutter ratio improvement. *IEEE Geoscience and Remote Sensing Letters*, 6(1): 23–27, 2009. doi: 10.1109/LGRS.2008.2006634. 3.2.2

[37] Alan J. Fenn, Peter T. Hurst, Joe Pacheco, Matthew Cornick, and Leonard I. Parad. Ultrawideband cavity-backed resistively loaded planar dipole array for ground penetrating radar. In *2013 IEEE International Symposium on Phased Array Systems and Technology*, pages 117–123, 2013. doi: 10.1109/ARRAY.2013.6731812. 4.2.1

[38] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. FlowNet: Learning optical flow with convolutional networks. *CoRR*, abs/1504.06852, 2015. URL http://arxiv.org/abs/1504.06852. 4.2.3, 4.5.3

[39] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. On-Manifold Preintegration for Real-Time Visual-Inertial Odometry. Technical report. URL https://youtu.be/CsJkci5lfco. 4.2.2, 4.2.2, 4.7.2, 4.7.2

[40] Paul Furgale, Tim Barfoot, and Nadeem Ghafoor. Rover-based surface and subsurface modeling for planetary exploration. In Andrew Howard, Karl Iagnemma, and Alonzo Kelly, editors, *Field and Service Robotics*, pages 499–508, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. ISBN 978-3-642-13408-1. 2.2.2

[41] Matthew Gadd, Daniele De Martini, and Paul Newman. Look Around You: Sequence-based Radar Place Recognition with Learned Rotational Invariance. In *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 2020. 2.3

[42] Paolo Gamba and Simone Lossani. Neural detection of pipe signatures in ground penetrating radar images. *IEEE Transactions on Geoscience and Remote Sensing*, 38(2):790–797, 2000. 5.2.3

[43] Katherine Grote, Taylor Crist, and Crystal Nickel. Experimental estimation of the GPR groundwave sampling depth. *Water Resources Research*, 46(10), 2010. doi: https://doi.org/10.1029/2009WR008403. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2009WR008403. (document), 2.3, 2.2.3

[44] Marco Hahn, Norbert Krüger, and Lehrstuhl Fur Kognitive Systeme. Junction detection and semantic interpretation using hough lines. In *International ICSC Symposium on Engineering of Intelligent Systems (EIS*, 2000. 5.2.1

[45] S. Hamran, T. Berger, S. Brovoll, L. Damsgård, Ø. Helleren, M. J. Øyan, H. E. Amundsen, L. Carter, R. Ghent, J. Kohler, M. Mellon, D. Paige, D. Plettemeier, and J. Eide. RIMFAX: A GPR for the Mars 2020 rover mission. In *2015 8th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, pages 1–4, 2015. doi: 10.1109/IWAGPR.2015.7292690. 2.2.2

[46] Herman Herman. *Robotic Subsurface Mapping Using Ground Penetrating Radar*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, May 1997. (document), 2, 3, 2.5, 2.6

[47] Herman Herman. Robotic Subsurface Mapping Using Ground Penetrating Radar. Technical report, 1997. 2.2.2

[48] Paul VC Hough. Method and means for recognizing complex patterns, December 18 1962. US Patent 3,069,654. 5.2.1

[49] Jamie H. Huang, J. L. Garry, G. E. Smith, and C. J. Baker. Array based passive radar target localization. In *2016 IEEE Radar Conference (RadarConf)*, pages 1–6, 2016. doi: 10.1109/RADAR.2016.7485106. 2.3

[50] Emanuel Huber and Guillaume Hans. RGPR — an open-source package to process and visualize GPR data. In *2018 17th International Conference on Ground Penetrating Radar (GPR)*, pages 1–4, 2018. doi: 10.1109/ICGPR.2018.8441658. 3.2.2

[51] Emanuel Huber and Guillaume Hans. RGPR — an open-source package to process and visualize gpr data. In *2018 17th International Conference on Ground Penetrating Radar (GPR)*, pages 1–4, 2018. doi: 10.1109/ICGPR.2018.8441658. 3.3

[52] Ruth Janning, T Horvath, A Busche, and L Schmidt-Thieme. Pipe localization by apex detection. In *IET International Conference on Radar Systems (Radar 2012)*, pages 1–6. IET, 2012. 5.2.3

[53] Ruth Janning, Tomáš Horváth, Andre Busche, and Lars Schmidt-Thieme. Gamrec: A clustering method using geometrical background knowledge for gpr data preprocessing. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 347–356. Springer, 2012. 5.2.3

[54] Ruth Janning, Andre Busche, Tomáš Horváth, and Lars Schmidt-Thieme. Buried pipe localization using an iterative geometric clustering on gpr data. *Artificial Intelligence Review*, 42(3):403–425, 2014. 5.2.3

[55] Harry M. Jol. In Harry M. Jol, editor, *Ground Penetrating Radar Theory and Applications*. Elsevier, Amsterdam, 2009. ISBN 978-0-444-53348-7. doi: https://doi.org/10.1016/B978-0-444-53348-7.00017-X. 3.2.2

[56] Michael Kaess. Simultaneous localization and mapping with infinite planes. *Proceedings - IEEE International Conference on Robotics and Automation*, 2015:4605–4611, 06 2015. doi: 10.1109/ICRA.2015.7139837. 6.2

[57] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John Leonard, and Frank Dellaert. iSAM2: Incremental smoothing and mapping using the Bayes tree. *Intl. J. of Robotics Research*, 31(2):216–235, February 2012. 2.1.3, 4.3.2, 4.7.4

[58] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960. 2.1.2

[59] Prannay Kaul, Daniele De Martini, Matthew Gadd, and Paul Newman. RSS-Net: Weakly-Supervised Multi-Class Semantic Segmentation with FMCW Radar. In *Proceedings of the IEEE Intelligent Vehicles Symposium (IV)*, 2020. 2.3

[60] Hans Kumar. Assessing the condition of buried pipe using ground penetrating radar.

Master's thesis, The University of Birmingham, Birmingham, UK, August 2013. 5.1

[61] Hans Kumar. Multi-session periodic slam for legged robots. Master's thesis, Carnegie Mellon University, Pittsburgh, PA, April 2021. 2.1.1

[62] H. M. La, R. S. Lim, B. Basily, N. Gucunski, J. Yi, A. Maher, F. A. Romero, and H. Parvardeh. Autonomous robotic system for high-efficiency non-destructive bridge deck inspection and evaluation. In *IEEE Intl. Conf. on Automation Science and Engineering (CASE)*, pages 1053–1058, 2013. doi: 10.1109/CoASE.2013.6653886. 2.2.2

[63] Jialong Lai, Yi Xu, Roberto Bugiolacchi, Xu Meng, Long Xiao, Minggang Xie, Bin Liu, Kaichang Di, Xiaoping Zhang, Bin Zhou, Shaoxiang Shen, and Luyuan Xu. First look by the Yutu-2 rover at the deep subsurface structure at the lunar farside. *Nature Communications*, 11(1):1–9, dec 2020. ISSN 20411723. doi: 10.1038/s41467-020-17262-w. 2.2.2

[64] Silvia Lameri, Federico Lombardi, Paolo Bestagini, Maurizio Lualdi, and Stefano Tubaro. Landmine detection from gpr data using convolutional neural networks. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 508–512, 2017. doi: 10.23919/EUSIPCO.2017.8081259. 5.2.3

[65] Wentai Lei, Feifei Hou, Jingchun Xi, Qianying Tan, Mengdi Xu, Xinyue Jiang, Gengye Liu, and Qingyuan Gu. Automatic hyperbola detection and fitting in gpr b-scan image. *Automation in Construction*, 106:102839, 2019. ISSN 0926-5805. doi: https://doi.org/10.1016/j.autcon.2019.102839. URL https://www.sciencedirect.com/science/article/pii/S0926580519301347. 5.2.3

[66] J. Lenarčič and F. Thomas. *Advances in Robot Kinematics: Theory and Applications*. Springer Netherlands, 2013. ISBN 9789401706575. URL https://books.google.com/books?id=zAPxCAAAQBAJ. 5.2.2, 6.2

[67] L. Likforman-Sulem, A. Hanimyan, and C. Faure. A hough based algorithm for extracting text lines in handwritten documents. In *Proceedings of 3rd International Conference on Document Analysis and Recognition*, volume 2, pages 774–777 vol.2, 1995. doi: 10.1109/ICDAR.1995.602017. 5.2.1

[68] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J. Leonard, David Cox, Peter Corke, and Michael J. Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016. doi: 10.1109/TRO.2015.2496823. 6.2

[69] Christian Maas and Jörg Schmalzl. Using pattern recognition to automatically localize reflection hyperbolas in data from ground penetrating radar. *Computers Geosciences*, 58:116–125, 2013. ISSN 0098-3004. doi: https://doi.org/10.1016/j.cageo.2013.04.012. URL https://www.sciencedirect.com/science/article/pii/S009830041300112X. 5.2.3

[70] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 Year, 1000km: The Oxford RobotCar Dataset. *The International Journal of Robotics Research (IJRR)*, 36 (1):3–15, 2017. doi: 10.1177/0278364916679498. URL http://dx.doi.org/10.1177/0278364916679498. 2.3

[71] John McCormac, Ronald Clark, Michael Bloesch, Andrew J. Davison, and Stefan Leutenegger. Fusion++: Volumetric object-level SLAM. *CoRR*, abs/1808.08378, 2018. URL http://arxiv.org/abs/1808.08378. 2.1.4

[72] John McDonald. Multi-session visual simultaneous localisation and mapping. December 2013. URL http://mural.maynoothuniversity.ie/4875/. 2.1.1

[73] Kevin Meier, Soon-Jo Chung, and Seth Hutchinson. Visual-inertial curve simultaneous localization and mapping: Creating a sparse structured world without feature points. *Journal of Field Robotics*, 35(4):516–544, June 2018. doi: https://doi.org/10.1002/rob.21759. URL https://resolver.caltech.edu/CaltechAUTHORS:20170518-135803615. 6.2

[74] Laurence Mertens, Raffaele Persico, Loredana Matera, and Sebastien Lambot. Automated detection of reflection hyperbolas in complex gpr images with no a priori knowledge on the medium. *IEEE Transactions on Geoscience and Remote Sensing*, 54(1):580–596, 2015. 5.2.3

[75] A.I. Mourikis and S.I. Roumeliotis. Analysis of positioning uncertainty in simultaneous localization and mapping (slam). In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 1, pages 13–20 vol.1, 2004. doi: 10.1109/IROS.2004.1389322. 2.1.1

[76] Raul Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *CoRR*, abs/1610.06475, 2016. (document), 2.1, 2.1.4

[77] Ian Nesbitt, François-Xavier Simon, Thomas Paulin, and teshaw. readgssi: an open-source tool to read and plot GSSI ground-penetrating radar data, May 2021. URL https://doi.org/10.5281/zenodo.4735817. 3.3

[78] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011. doi: 10.1109/ISMAR.2011.6092378. 2.1.4

[79] Kenzo Nonami. *Development of Mine Detection Robot Mine Hunter Vehicle (MHV), Controlled Metal Detector and Multi-functional Hydraulic Manipulator*, pages 123–143. Springer London, London, 2009. ISBN 978-1-84882-346-4. doi: 10.1007/978-1-84882-346-4_8. 2.2.2

[80] Teddy Ort, Igor Gilitschenski, and Daniela Rus. Autonomous navigation in inclement weather based on a localizing ground penetrating radar. *IEEE Robotics and Automation Letters*, 5(2):3267–3274, 2020. ISSN 23773766. doi: 10.1109/LRA.2020.2976310. (document), 1.1, 4.2.1, 4.3, 4.4, 4.2.1, 4.5, 6.1

[81] Teddy Ort, Igor Gilitschenski, and Daniela Rus. GROUNDED: The Localizing Ground Penetrating Radar Evaluation Dataset. In *Proceedings of Robotics: Science and Systems*, Virtual, July 2021. doi: 10.15607/RSS.2021.XVII.080. (document), 3.3, 4.2.1, 6.2

[82] Surf Pacific. Avoid close calls: Underground service locator: Provac australia, Jul 2021. URL https://www.provac.net.au/avoid-close-calls-with-an-underground-service-locator/. (document), 5.1

[83] Yeong Sang Park, Jinyong Jeong, Youngsik Shin, and Ayoung Kim. Radar dataset for robust localization and mapping in urban environment. In *ICRA Workshop on Dataset Generation and Benchmarking of SLAM Algorithms for Robotics and VR/AR*, 2019. 2.3

[84] Yeong Sang Park, Joowan Kim, and Ayoung Kim. Radar localization and mapping for indoor disaster environments via multi-modal registration to prior lidar map. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1307–1314, 2019. doi: 10.1109/IROS40897.2019.8967633. 2.3

[85] Yeong Sang Park, Young-Sik Shin, and Ayoung Kim. Pharao: Direct radar odometry using phase correlation. In *2020 IEEE International Conference on Robotics and Automation, ICRA 2020, Paris, France, May 31 - August 31, 2020*, pages 2617–2623. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9197231. URL https://doi.org/10.1109/ICRA40945.2020.9197231. 2.3

[86] Edoardo Pasolli, Farid Melgani, and Massimo Donelli. Automatic analysis of gpr images: A pattern-recognition approach. *IEEE Transactions on Geoscience and Remote Sensing*, 47(7):2206–2217, 2009. 5.2.3

[87] Aleksandar Ristić, Željko Bugarinović, Miro Govedarica, Lara Pajewski, and Xavier Derobert. Verification of algorithm for point extraction from hyperbolic reflections in gpr data. In *2017 9th International Workshop on Advanced Ground Penetrating Radar (IWAGPR)*, pages 1–5, 2017. doi: 10.1109/IWAGPR.2017.7996109. 5.2.3

[88] Stefan Saftescu, Matthew Gadd, Daniele De Martini, Dan Barnes, and Paul Newman. Kidnapped Radar: Topological Radar Localisation using Rotationally-Invariant Metric Learning. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020. 2.3

[89] João Machado Santos, Tomáš Krajník, and Tom Duckett. Spatio-temporal exploration strategies for long-term autonomy of mobile robots. *Robotics and Autonomous Systems*, 88:116–126, 2017. ISSN 0921-8890. doi: https://doi.org/10.1016/j.robot.2016.11.016. URL https://www.sciencedirect.com/science/article/pii/S092188901630690X. 6.2

[90] Harsha M. Sathyendra and Bryan D. Stephan. Automated radar image target recognition based on segmented feature extraction using adaptive length estimation and hough lines aiding optimized neural network classification. In *2013 IEEE Radar Conference (RadarCon13)*, pages 1–5, 2013. doi: 10.1109/RADAR.2013.6585986. 5.2.1

[91] Motoyuki Sato. *Principles of Mine Detection by Ground-penetrating Radar*, pages 19–26. Springer London, London, 2009. ISBN 978-1-84882-346-4. doi: 10.1007/978-1-84882-346-4_2. 2.2.2

[92] Richard A. Schultz. *Geologic Fracture Mechanics*. Cambridge University Press, 2019. doi: 10.1017/9781316996737. 1.2

[93] Marcel Sheeny, Emanuele De Pellegrin, Saptarshi Mukherjee, Alireza Ahrabian, Sen Wang, and Andrew Wallace. Radiate: A radar dataset for automotive perception. *arXiv*, 2020. 2.3

[94] Solomon Eyal Shimony. Finding maps for belief networks is np-hard. *Artificial Intelligence*, 68(2):399–410, 1994. ISSN 0004-3702. doi: https://doi.org/10.1016/0004-3702(94)90072-8. URL https://www.sciencedirect.com/science/article/pii/0004370294900728. 2.1.2

[95] M.I. Skolnik. *Radar Handbook, Third Edition*. Electronics electrical engineering. McGraw-Hill Education, 2008. ISBN 9780071485470. URL https://books.google.com/books?id=76uF2Xebm-gC. 2.2.1, 2.2.3

[96] Randall Smith, Matthew Self, and Peter Cheeseman. *Estimating Uncertain Spatial Relationships in Robotics*, pages 167–193. Springer New York, New York, NY, 1990. ISBN 978-1-4613-8997-2. doi: 10.1007/978-1-4613-8997-2_14. URL https://doi.org/10.1007/978-1-4613-8997-2_14. 2.1.2

[97] Paloma Sodhi, Michael Kaess, Mustafa Mukadam, and Stuart Anderson. Learning tactile models for factor graph-based estimation. *arXiv preprint arXiv:2012.03768*, 2020. 4.2.3, 4.3.1

[98] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, Oct. 2012. 4.8.4

[99] Sudharshan Suresh. Localization and active exploration in indoor underwater environments. Master's thesis, Carnegie Mellon University, Pittsburgh, PA, August 2019. 2.1.2

[100] Tim Y. Tang, Daniele De Martini, Shangzhe Wu, and Paul Newman. Self-supervised localisation between range sensors and overhead imagery. In *Robotics: Science and Systems (RSS)*, 2020. 2.3

[101] Alan Taylor. One year on mars: The curiosity rover. https://www.theatlantic.com/photo/2013/08/one-year-on-mars-the-curiosity-rover/100567/. (document), 1.2

[102] S. Thrun, W. Burgard, D. Fox, and R.C. Arkin. *Probabilistic Robotics*. Intelligent Robotics and Autonomous Agents series. MIT Press, 2005. ISBN 9780262201629. URL https://books.google.com/books?id=2Zn6AQAAQBAJ. 2.1.1

[103] E.C. Utsi. *Ground Penetrating Radar: Theory and Practice*. Elsevier Science, 2017. ISBN 9780081022177. URL https://books.google.com/books?id=JSK2DQAAQBAJ. 2.2.1

[104] Craig Warren, Antonios Giannopoulos, and Iraklis Giannakis. gprmax: Open source software to simulate electromagnetic wave propagation for ground penetrating radar. *Computer Physics Communications*, 209:163–170, 2016. ISSN 0010-4655. doi: https://doi.org/10.1016/j.cpc.2016.08.020. URL https://www.sciencedirect.com/science/article/pii/S0010465516302533. 3.3

[105] Rob Weston, Sarah Cen, Paul Newman, and Ingmar Posner. Probably unknown: Deep inverse sensor modelling radar. 2019. 2.3

[106] William Whittaker. 3d mine mapping. https://www.ri.cmu.edu/project/3d-mine-mapping/. (document), 1.2

[107] R. M. Williams, L. E. Ray, J. H. Lever, and A. M. Burzynski. Crevasse detection in ice sheets using ground penetrating radar and machine learning. *IEEE J. of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(12):4836–4848, 2014. doi: 10.1109/JSTARS.2014.2332872. 2.2.2

[108] Oliver J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, August 2007. URL https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf. 4.2.2

[109] Jingya Yan, Siow Wei Jaw, Kean Huat Soon, Andreas Wieser, and Gerhard Schrotter. Towards an underground utilities 3d data model for land administration. *Remote Sensing*, 11(17), 2019. ISSN 2072-4292. doi: 10.3390/rs11171957. URL https://www.mdpi.com/2072-4292/11/17/1957. 1.2

[110] Shichao Yang, Yu Song, Michael Kaess, and Sebastian Scherer. Pop-up slam: Semantic monocular plane slam for low-texture environments, 2017. 6.2

[111] Yulin Yang and Guoquan Huang. Aided inertial navigation: Unified feature representations and observability analysis. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3528–3534, 2019. doi: 10.1109/ICRA.2019.8793507. 5.2.2

[112] Yulin Yang, Patrick Geneva, Kevin Eckenhoff, and Guoquan Huang. Visual-inertial odometry with point and line features. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2447–2454, 2019. doi: 10.1109/IROS40897.2019.8967905. 5.2.2, 6.2

[113] Brent Yi, Michelle A. Lee, Alina Kloss, Roberto Martín-Martín, and Jeannette Bohg. Differentiable factor graph optimization for learning smoothers, 2021. 4.5.3

[114] Ji Zhang and S. Singh. LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems (RSS)*, 2014. 2.1.4