# Robust and Efficient Semantic Sensor Registration for Mobile Robotics in Unorganized, Natural, Scenes

by

Steven A. Parkison

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2020

Doctoral Committee:

Professor Ryan M. Eustice, Co-Chair
Associate Professor Odest C. Jenkins, Co-Chair
Professor Jason J. Corso
Professor Jessy Grizzle
Associate Professor Matthew Johnson-Roberson

Steven A. Parkison

sparki@umich.edu

ORCID iD: 0000-0001-8265-3555

# ACKNOWLEDGMENTS

In a lot of ways, this thesis is the collaborative work of dozens of people in addition to me. While this reflects the nature of research, I would like to take a moment to thank a few of the many people that helped me during my dissertation work.

First, I want to thank my committee. The feedback I got during and after my proposal presentation was invaluable, and I appreciate everyone accommodating the tight scheduling requirements for our meetings. That goes double for the co-chairs of my committee. Ryan, I am thankful for the opportunities and resources you provided. Chad, I appreciate your willingness to always provide guidance and insights, even when I unexpectedly ambushed you.

I would also like to thank all my fellow research assistants who help me during my thesis work. To all the PeRL lab members (Gaurav, GT, Nick, Enric, Jeff, Paul, Vittorio, Wolcott, Stephen, Steve, Alex, Derrick, Jie, Arash, Josh, Lu, Maani, James, and Ray. Sorry for messing up your pictures during the defense presentation.), I'd like to thank you all for showing me the ropes and helping me get my sea legs. I hope you all do well in your future endeavors. I would also like to thank the members of the Laboratory for Progress and the DROP lab for letting me bounce ideas off them and providing feedback.

There are also a lot of people in the EECS department, Robotics Institute, and wider College of Engineering community at the University of Michigan that I should also thank. There are countless things I learned while taking classes that have been invaluable but don't make an appearance in this thesis.

To my family, thank you for putting up with me for the past few years. I appreciate the support you have provided despite me having been constantly busy and distant. I hope to make up for it in the future.

To all my friends, you are great. Thanks to all the co-inhabitants of 334 East Kingsley. You provided a stress free place to live. But also support, interesting conversations, and entertainment, which helped me keep my sanity during the work towards a PhD. And thanks to everyone who has gone on bike rides with me, the rides are always easier when there is someone else there to help break the wind (something that is as true in life as it is in the saddle). And to Audrey, I am thankful I had you to commiserate with the last few years. Thanks for all the support through this process, I hope to repay it as you finish your PhD.

And Finally, I should probably thank my funding providers. Thank you TRW and Ford for

supporting my work in the early years. The NSF GRFP was invaluable for allowing me the flexibility to research what I found interesting in the middle of my PhD. And thank you to the Toyota Research Institute for funding my final year.

# TABLE OF CONTENTS

**Acknowledgments** . . . . . **ii**

**List of Figures** . . . . . **vi**

**List of Tables** . . . . . **viii**

**List of Appendices** . . . . . **ix**

**List of Acronyms** . . . . . **x**

**Abstract** . . . . . **xii**

**Chapter**

**1 Introduction** . . . . . **1**

1.1 3D Sensor Registration . . . . . 2
1.1.1 Types of 3D Sensors . . . . . 2
1.1.2 Introduction to the Registration Problem . . . . . 3
1.1.3 Registration Algorithms . . . . . 4
1.1.4 Application of Sensor Registration . . . . . 8
1.2 Semantic Classification . . . . . 9
1.3 Semantics and Geometry . . . . . 11
1.4 Datasets for Semantics and Geometry . . . . . 15
1.4.1 KITTI Dataset . . . . . 16
1.4.2 TUM RGB-D Dataset . . . . . 16
1.4.3 SceneNet RGB-D Dataset . . . . . 19
1.5 Thesis Outline . . . . . 19
1.5.1 Document Roadmap . . . . . 21

**2 Semantic Iterative Closest Point Through Expectation Maximization** . . . . . **24**

2.1 Introduction . . . . . 24
2.2 Related Work . . . . . 25
2.3 Problem Statement and Formulation . . . . . 28
2.4 Generalized ICP on $\mathrm{SE}(3)$ . . . . . 29
2.5 Semantic Iterative Closest Point . . . . . 31
2.6 Evaluation . . . . . 33
2.6.1 Optimization . . . . . 34
2.6.2 KITTI Visual Odometry Dataset . . . . . 34

2.6.3 SceneNet RGBD Dataset . . . 37
2.7 Conclusion . . . 40

**3 Boosting Shape Registration Algorithms via Reproducing Kernel Hilbert Space Regularizers . . . 43**

3.1 Introduction . . . 43
3.1.1 Contributions . . . 46
3.1.2 Outline . . . 47
3.1.3 Representation and Reproducing Kernel Hilbert Space . . . 47
3.2 Problem Statement and Formulation . . . 48
3.3 A Class of $\mathcal{H}_k$-Regularized Shape Registration Algorithms . . . 49
3.3.1 $\mathcal{H}_k$-Regularization via Sparse Bayesian Inference . . . 50
3.3.2 Algorithmic Implementation . . . 52
3.4 Experimental Results . . . 52
3.4.1 LIDAR: KITTI Odometry dataset . . . 55
3.4.2 RGB-D: TUM RGB-D SLAM dataset . . . 59
3.5 Conclusion . . . 60

**4 2D to 3D Line-Based Registration with Unknown Associations via Mixed-Integer Programming . . . 63**

4.1 Introduction . . . 63
4.2 Problem Formulation . . . 65
4.2.1 Plücker Coordinates . . . 66
4.2.2 Line-Based Registration . . . 66
4.3 Method . . . 69
4.3.1 Mixed-Integer Formulation . . . 69
4.3.2 Constraining the Problem to $\mathrm{SE}(2)$ . . . 70
4.3.3 Fitting to a Valid Transformation . . . 71
4.4 Evaluation . . . 71
4.4.1 VGG Multiview Dataset . . . 71
4.4.2 Autonomous Vehicle Dataset . . . 75
4.5 Conclusions . . . 78

**5 Conclusion . . . 82**

5.1 Contributions . . . 82
5.2 Future Work . . . 83
5.2.1 GPU Accelerated Algorithms . . . 83
5.2.2 Curvature for Semantic Registration . . . 87
5.2.3 Dynamic Scenes . . . 93
5.2.4 Eliminating Data Association Challenges . . . 94
5.2.5 Bespoke Optimization Approaches . . . 95

**Appendices . . . 100**

**Bibliography . . . 109**

# LIST OF FIGURES

**FIGURE**


1.1 A comparison of two 3D sensors . . . 3
1.2 Different modalitits of range sensors . . . 4
1.3 Illustration of the iterative closest point (ICP) algorithm . . . 5
1.4 Generalized Iterative closest point (ICP) algorithm . . . 6
1.5 Illustration of the KD tree splitting procedure . . . 7
1.6 Illustration of projective association . . . 8
1.7 An Example CNN architecture. . . . 10
1.8 Sample annotations from the MS COCO dataset . . . 10
1.9 Annotated Cityscapes images . . . 12
1.10 Dense semantic map . . . 13
1.11 Illustration of Semantic Classification on 3D Data . . . 14
1.12 Sample map generated from a LIDAR sequence in the KITTI odometry dataset . . . 17
1.13 Data from the TUM RGB-D dataset . . . 18
1.14 Ground truth data from the SceneNet RGB-D dataset . . . 19

2.1 Sematic registration results from SceneNet RGBD . . . 26
2.2 Convergenc evaluation for GICP and GICP-$\mathrm{SE3}$ . . . 30
2.3 CDF and box plots for KITTI Sequence 05 . . . 36
2.4 Registered point clouds using Semantic ICP and GICP . . . 38
2.5 Initial alignment vs. final alignment for registration alogrithms on the KITTI dataset . . . 39
2.6 CDF and box plots for SceneNet RGB-D dataset . . . 41

3.1 An illustration of the proposed regularization method . . . 44
3.2 Trajectory results of the proposed method versus benchmark algirhtms . . . 53
3.3 A detailed view of sequence 00 reconstructed using $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ odometry . . . 54
3.4 A detailed view of sequence 01 reconstructed using $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ odometry . . . 54
3.5 Average translation and rotation error vs speed on the KITTI Odometry dataset. . . . 56
3.6 Scatter plots of initial versus final error of the various methods on KITTI Odometry . . . 58
3.7 Timing comparison for the various algorithms on the KITTI Odometry dataset . . . 59
3.8 CDF plots for translational error from the TUM dataset . . . 60

4.1 Alignment of 3D lines with 2D lines using the proposed approach. . . . 64
4.2 Complexity of data association. . . . 68
4.3 Results on the Corridor sequence with known associations. . . . 72
4.4 Results on the Merton sequence with known associations. . . . 73

4.5 Results on the Wadham sequence with known associations. . . . . 73
4.6 Results on the Library sequence with known associations. . . . . 74
4.7 VGG Results without known associations. . . . . 75
4.8 Example result of VGG data without known association . . . . . 76
4.9 Illustration of the line extraction procedure used for the driving data. . . . . 77
4.10 Frame A result using data collected from an autonomous vehicle platform. . . . . 79
4.11 Frame B result using data collected from an autonomous vehicle platform. . . . . 80
4.12 Frame C result using data collected from an autonomous vehicle platform. . . . . 81

5.1 Growth in the number of processing unit per chip in the last decade . . . . . 84
5.2 Runtime and speed up of GPU registration conponets . . . . . 86
5.3 Runtime and Speed up of GPU GICP Algorithm . . . . . 87
5.4 Comparison of the effects of using CPU and GPU registration algorithms for SLAM . 88
5.5 Illustration of surface in $\mathbb{R}^3$ . . . . . 89
5.6 Illustration of a point on a surface . . . . . 89
5.7 Illustration of a normal of surface . . . . . 90
5.8 Illustration of a curve on a surface . . . . . 91
5.9 Curvature computed on LIDAR data from the KITTI odometry dataset . . . . . 93
5.10 Illustration of bounding a point in an image . . . . . 96
5.11 Comparison of analytical bounds and numeric sampling over transformation set . . . . 98
5.12 Illustration of a sparse table . . . . . 99

# LIST OF TABLES

**TABLE**


2.1 Paramters used for optimizatioin in Semantic ICP evaluation . . . . . 34
2.2 Dilation CNN performance measure on the KITTI Odomentry Dataset . . . . . 35
2.3 KITTI odometry dataset results, error metrics and runtime. . . . . 35
2.4 DeepLab-ResNet performance measure on the SceneNet RGBD Dataset train_0 . . . . 40
2.5 SceneNet RGB-D accuracy and runtime results, semantic ICP vs benchmarks. . . . . 40

3.1 Parameters used for evaluating $\mathcal{H}_k$-GICP algorihm . . . . . 52
3.2 Results of the evaluation of $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ using the KITTI odometry benchmark . . 55
3.3 Evaluation of $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ on the TUM RGB-D SLAM dataset . . . . . 61

4.1 Results on the VGG Multiview dataset with known associations . . . . . 74
4.2 Results on the driving dataset with unknown associations. . . . . 78

5.1 Runtime of curvature estimation . . . . . 92

# LIST OF APPENDICES

**APPENDIX**

**A Lie Group Notations** . . . . . **101**

**B Manifold Optimization** . . . . . **103**

**C Mixed-Integer Programming** . . . . . **105**

**D Software Repositories** . . . . . **107**

# LIST OF ACRONYMS

**2D** two-dimensional

**3D** three-dimensional

**BnB** branch and bound

**CNN** Convolutional Neural Networks

**CPU** central processing unit

**CUDA** Compute Unified Device Architecture

**EM** expectation-maximization

**GICP** Generalized Iterative Closest Point

**GPS** global positioning system

**GPU** graphics processing unit

**ICP** iterative closest point

**IMU** inertial measurement unit

**INS** inertial navigation system

**IR** infrared

**KD** $k$-dimensional

**LIDAR** light detection and ranging

**MAV** micro-aerial vehicles

**MIP** mixed-integer program

**NDT** normal distributions transform

**PnP** Perspective-n-Point

**PnL** Perspective-n-Line

**RANSAC** random sample consensus

**RGB-D** red, gleen, blue, and depth

**SIFT** scale-invariant feature transform

**SIMD** single input, multiple data

**SLAM** simultaneous localization and mapping

**SVD** singular value decomposition

# ABSTRACT

Advances in sensing and computing hardware have led to renewed interest in registration algorithms. In particular, the proliferation of 3D light detection and ranging (LIDAR) sensors and RGBD cameras and their use in robotic systems require efficient, robust, and accurate estimation algorithms for use in mapping, localization, and tracking tasks. Most modern approaches to autonomous driving require localizing and calibrating multiple LIDAR sensors, both of which are registration tasks. Meanwhile, tasks in the domain of indoor robotics require both localizing the robot and localizing objects of interest in the environment. The registration problem is that of trying to find the rigid body transformation between two measurements. This can include consecutive measurements (producing an odometry estimate), measurements from disparate points in time (such as for localization and mapping), and between different sensors (such as for calibrating multiple sensors on a platform).

Semantic detection and segmentation have similarly significantly progressed. Semantic inference on images and point clouds has shown increasing value in vision-based applications. The application of Convolutional Neural Networks (CNNs) has improved the computational efficiency of semantic segmentation techniques with superior performance in both indoor and outdoor benchmarks. Together with pose estimation techniques, multiple scenes can be segmented and combined to perform semantic mapping or object tracking; nevertheless, most semantic mapping and object tracking research has focused on performing pose estimation, and then semantic inference. So far, most research has not focused on joint semantic and metric estimation.

This thesis focuses on leveraging semantic inference to enable efficient and robust sensor registration. In robotics, semantic inference is increasingly used for downstream reasoning tasks. This thesis explores how that inference can be used in upstream task such as egomotion estimation, object pose estimation, and multisensor calibration. This work is based on improving the Iterative Closest Point (ICP) algorithm.

Our first contribution in this thesis explores how probabilistic semantic labels can be used in sensor registration. We present an approach that uses the Expectation Maximization (EM) technique to improve associations in the ICP framework. We also use an M-Estimator and optimize directly on the $\mathbf{SE}(3)$ manifold to improve the robustness. Our results on publicly available indoor and outdoor data sets show that semantics can help improve registration accuracy. For the second

contribution, we add informative channels to the ICP framework to aid in object-level registration. This includes work on using sparse kernels to represent intensity and color channels for regularizing the registration problem, and work on curvature based alignment to improve object pose estimation. This technique extends registration algorithms beyond their purely geometric base. Our third contribution is a reformulation the registration problem as a mixed integer program (MIP). Most previous approaches to sensor registration use gradient-based optimization techniques. If the cost function used is nonconvex, they are prone to getting caught in local minima. The problem is reformulated as a MIP by linearizing the cost function and representing the data association as an integer valued variable.

This thesis focuses on developing robust and accurate registration techniques for mobile robotic applications. It presents results and proposed evaluation in the areas of indoor home robotics and autonomous driving, many of which are publicly available benchmark data sets. Sensor registration is a fundamental component of many robotic systems, and the advances proposed in this thesis have the potential to benefit many more aspects of perceptual systems.

# CHAPTER 1

# Introduction

Individuals and organizations are increasingly using mobile robots to perform tasks in real-world environments. Such tasks include low speed shuttles to provide transportation on a preset path and vacuum cleaners that autonomously clean an area. These systems function using a metric representation of the world around them. To move beyond simple automation, robotic systems need a higher reasoning that is based upon both precise metric knowledge and a semantic understanding of what makes up the environment.

In part, new hardware with better capabilities enabled the recent growth of mobile robotics. This includes sensors capable of high throughput 3D measurements such as light detection and ranging (LIDAR) sensors and RGB-D sensors. It also includes improvements in computing hardware. Higher thread count CPUs enabled multiple processes to run at once and the fine grain parallelism of GPUs allowed for the speedup of massive vector operations. These improvements have enabled algorithms that provided real-time and dense models of an environment that are needed for many mobile robotics tasks. Up to this point, most algorithms and methods focus on either metric or semantic models. Yet, mobile robots could benefit from concurrent metric and semantic representations (i.e. joint representation) to perform complex tasks in the world. The field of mobile robotics needs better joint algorithms to enable higher reasoning. From navigating complex environments to assistive home robotics, higher reasoning is required develop more advanced automation.

Toward this goal, this thesis focuses on improving the state-of-the-art in joint metric/semantic 3D sensor registration. These approaches to registration enable more accurate joint scene estimation for use by down stream robotic tasks. We first propose a joint framework for sensor registration that improves both geometric correspondence and semantic consistency. Next we propose methods that go beyond traditional point to point correspondences to further improve joint registration. Finally, we propose a new multi-resolution search approach for optimizing our joint registration cost function. We demonstrate these proposed approaches on real-world pubic datasets collected on robotic platforms.

The rest of this chapter is arranged as follows. We first present the sensor registration problem in Section 1.1, and review literature in this area. We then, in Section 1.2, present the current role of semantics in mobile robotics research, and then in Section 1.3 we present recent work in combining geometry and semantics. In Section 1.4 we introduce some publicly available datasets which combine robotic systems and semantic that we used to evaluate the methods presented in this document. Finally, in Section 1.5, we present the outline of the remainder of this thesis.

## 1.1 3D Sensor Registration

As the term is used in this thesis, sensor registration is a class of measurement algorithms that calculate the $\mathrm{SE}(3)$ transformation between multiple observations, or between an observation and a prior model. Such computation can be performed on a variety of perceptive sensors, but for this thesis we mostly focus on a subset of sensor that measure three-dimensional (3D) points in their environments. These algorithms play an important role in mapping, odometry, and object localization tasks performed by mobile robotic systems.

### 1.1.1 Types of 3D Sensors

3D sensors are the subset of perceptual sensors whose observations can measure direction and ranges in 3D space. This includes LIDAR systems that consist of spinning beams which produce planar range measurements, and red, gleen, blue, and depth (RGB-D) sensors that are geometrically similar to cameras and measure "depth images" either via structured light or time-of-flight observations. Both modalities are active sensors that project infrared (IR) light into the environment and measure how it returns (IR is used due to its closeness to the visual spectrum without affecting passive visual measurements). Some sensors can lose their ability to detect depth outside or in the presence of another sensor due to interference from other IR sensors. This is most common for RGB-D sensors that use structured light. Examples of each sensor are shown in Figure 1.1.

Because they rely on measuring light that has traveled from the sensor, to its surroundings, and back again, active 3D sensor only measure the position of surfaces in their immediate surroundings, and only a subset of those surfaces that have an unobstructed line-of-sight to the sensor. The differing geometries of these sensors lead to differing amounts of point density. LIDAR sensors have a wide field of view, sometimes up to 360°in the yaw axis, but can have sparse density in some dimensions. This has led to them being popular for egomotion estimation and large object tracking. RGB-D sensors generally have a smaller field of view, but are much denser where they do get observations. As the name suggests, they also passively observe the RGB light spectrum


(a) Intel RealSense SR305


(b) Velodyne Puck

Figure 1.1: Two 3D sensors. The Intel RealSense is a structured light RGB-D sensor designed for indoor tasks. The Velodyne Puck is a 32 beam LIDAR sensor.

magnitudes associated with their depth value. This makes RGB-D sensors useful for detecting and classifying parts of a scene. The differing geometries of these sensors is illustrated in Figure 1.2.

### 1.1.2 Introduction to the Registration Problem

The set of observations collected by these range sensors can be organized into point clouds, $\mathbf{x}_i \in \mathcal{X}$ where $\mathcal{X} \subset \mathbb{R}^3$, sometimes referred to as point sets. The point cloud (or point set) registration problem is usually formulated as finding the rigid body transformation in $\mathrm{SE}(3)$ that transforms one point cloud, nominally called the source point cloud $\mathcal{X}_s$, into the reference frame of another point cloud, called the target point cloud $\mathcal{X}_t$. There are two broad classes of registration algorithms. Coarse alignment algorithms generally make few assumptions about the amount of overlap between observation and focus on detecting crude alignments for things such as loop closures. Fine alignment is the focus of this thesis and is the class of algorithms that are concerned with finding accurate transformations between observations that are assumed to have a non negligible amount of overlap. This registration problem is commonly represented as the following optimization

$$\arg\min_{\mathbf{T}} \sum_i ||\mathbf{x}_i^t - \mathbf{T}\{\mathbf{x}_i^s\}|| \tag{1.1}$$

where $\mathbf{T}$ is the $\mathrm{SE}(3)$ action on a point, or

$$\mathbf{T}\{\mathbf{x}\} = \mathbf{R}\mathbf{x} + \mathbf{p} \tag{1.2}$$

where $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{p} \in \mathbb{R}^3$.

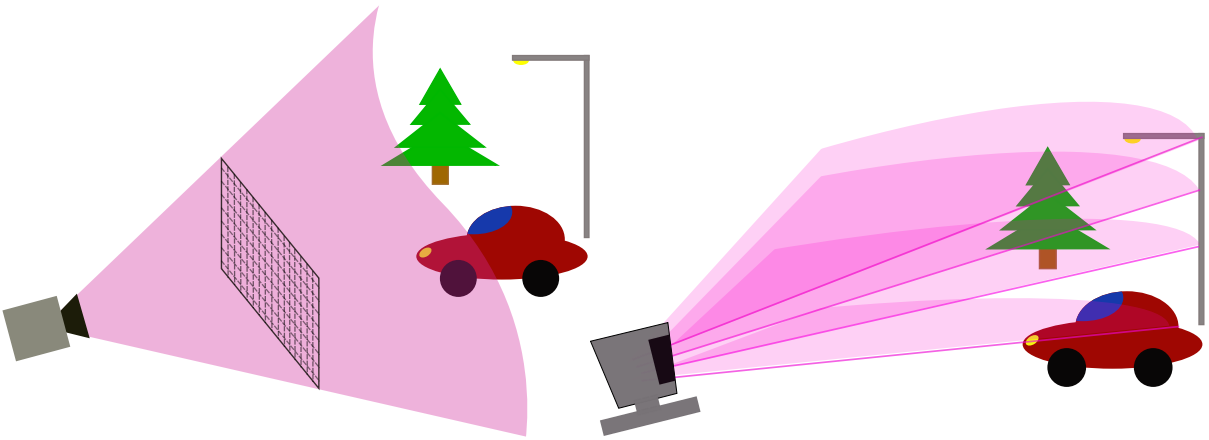


Figure 1.2: Illustration of the differences in geometry between the modalities of range sensors. RGB-D sensors produce dense observations in a limited field of view. LIDAR sensors produce observations with a wide horizontal field of view but with limited vertical density (in the frame of the sensor).

### 1.1.3 Registration Algorithms

An iterative method called iterative closest point (ICP) was first proposed by Besl and McKay (1992). They present an iterative two step procedure that alternates between

1. Finding associations between the target and source points.

2. Minimizing the distance between those associations.

and repeats until some convergence parameter is met. The second step represents the problem presented in equation 1.1. Underlying that equations is that it needs information on which points in each point cloud are observing the same spot in the environment. This represents a latent variable in the minimization problem, in that LIDAR and RGB-D sensors do not observe associations between point clouds. The formulation presented by Besl and McKay (1992) approximates this association by performing a nearest neighbor search for every source point cloud in the target point cloud.

There has also been work on alternative cost functions using other geometric primitives, such as point-to-line distance (Censi (2008)) and point-to-plane distance (Chen and Medioni (1991a)). Cost functions that have been shown to improve convergence speed and accuracy, by both Rusinkiewicz and Levoy (2001) and Pomerleau et al. (2013). Biber and Straßer (2003) define normal distributions using points in the target point cloud that fall into voxels of the environment. The objective function is defined as the probability that a point in the source point cloud is within the distributions of the target point cloud.

A more recent adaptation is the basis for some of the work presented in this thesis. By fitting a Gaussian distribution to each point, the Generalized Iterative Closest Point (GICP) algorithm by

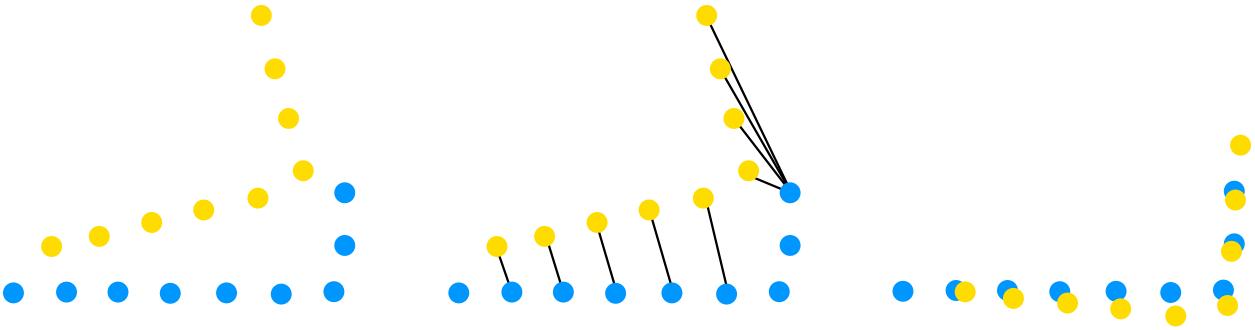


Figure 1.3: Illustration of the ICP algorithm, showing how new associations between source (yellow) and target (blue) points are found at each iteration, and then the distance between them is minimized.

Segal, Haehnel, and Thrun (2009) is more robust to sensor noise. GICP also accounts for the fact that 3D range sensors measure points on surfaces in the environment. These surfaces when viewed at a small enough level can be considered locally planar. GICP modifies the smallest eigenvalue of the sample covariance to be some small $\epsilon$ value to approximate this, as illustrated in Figure 1.4.

#### 1.1.3.1 Data Association

Both ICP and GICP perform nearest neighbor association using a $k$-dimensional (KD) tree data structure. A KD tree is a balanced binary tree that allows for quick nearest neighbor queries. On average they are $\mathcal{O}(\log n)$ and worst case is $\mathcal{O}(n)$. In the ICP algorithm, nearest neighbor searches are used to determine associations to compute residuals. For the GICP a nearest neighbor query is used for associations and to compute the local covariance used to represent the local plane of the surface being measured. The trade off is that KD trees require initial time to construct the data structure. Depending on the approach taken, building the data structure can take up to $\mathcal{O}(n \log n)$.

In contrast, another approach is to use the geometry of the sensor to compute the associations. This is called projective association and is most common with RGB-D sensors, an example of which was presented by Kerl, Sturm, and Cremers (2013). These methods are sometimes described projective association. Using the camera calibration matrix $K$ and current estimated transformation, the source point cloud is transformed and the projected into the target point cloud image. The pixel each source point is projected onto, represents the target point it is association with. With this approach associations can be found in $\mathcal{O}(c)$ (constant) time with no data structure to build. The trade- off with this approach is that occlusions can cause mis-associations, so these approaches are best used when the transformation between target and source clouds is minimal and therefor there are few occlusions. This method also works better with higher data density which is why it is more commonly used with RGB-D sensors and not LIDAR sensors. These approaches are sometimes

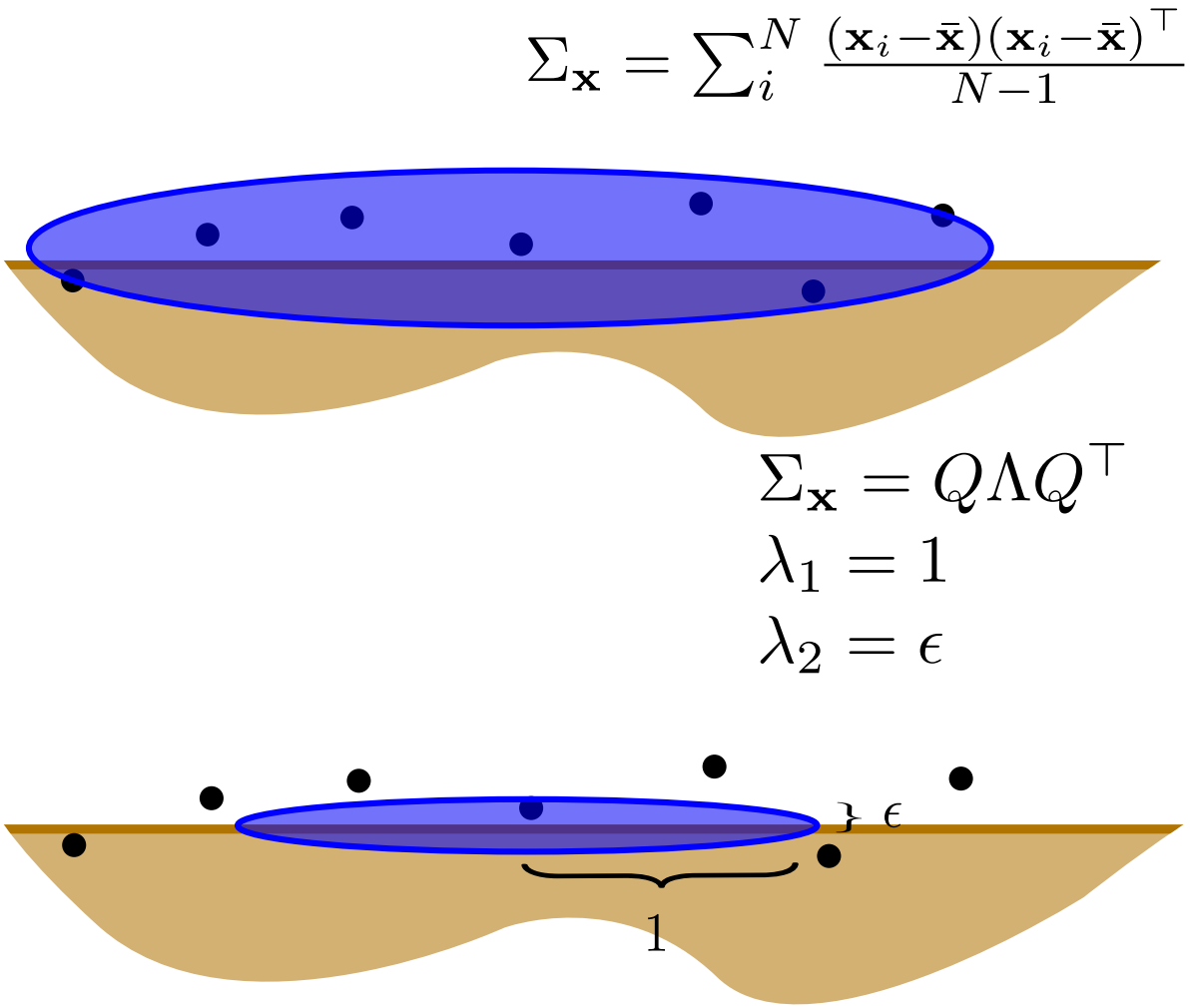



Figure 1.4: Illustration of the covariance fitting procedure used in the Generalized ICP algorithm, showing how the eigenvalues of the sample covariance matrix are adjusted to approximate a planar surface.

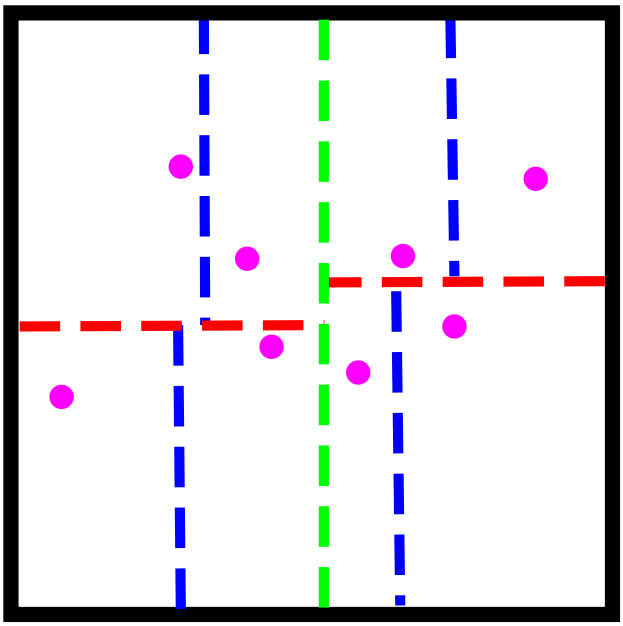


Figure 1.5: Illustration of the KD tree partitioning procedure of a set of points in 2D. The KD tree data structures recursively splits the set in half creating a balanced tree with $\mathcal{O}(\log n)$ average search complexity. In this illustration, the top level partition is green, the next level of partitions are in red, and the final set of partitions are in blue.

described as maximizing photo-consistency.

#### 1.1.3.2 Residual Minimization

Gradient-based approaches are the most common way to minimize the residuals of the registration problem as presented in (1.1). Besl and McKay (1992) presented a line search approach to solving the ICP problem. The orignal implementation of GICP used a version of the Broyden–Fletcher–Goldfarb–Shanno algorithm, a quasi-Newton method where the Hessian matrix is approximated using updates to the gradient. Kerl, Sturm, and Cremers (2013) use an iteratively reweighted least squares approach to maximizing their photo-consistency objective.

Gradient-based approaches are usually fast at the residual minimization step of registration. The trade of, is that for non-convex formulations of the registration problem, gradient-based approaches are susceptible to converging to local minimum. In constrast branch and bound (BnB) works by iteratively evaluating a set of possible transformations. It starts with the full set of possible transformations, and splits and bounds the intervals to find the globally optimal parameter. The bounds allows for searching only the intervals that have a possibility to improve the current best parameter, reducing the number of evaluations that are needed.

Others have explored branch and bound search for sensor registration. Yang, Li, and Jia (2013)

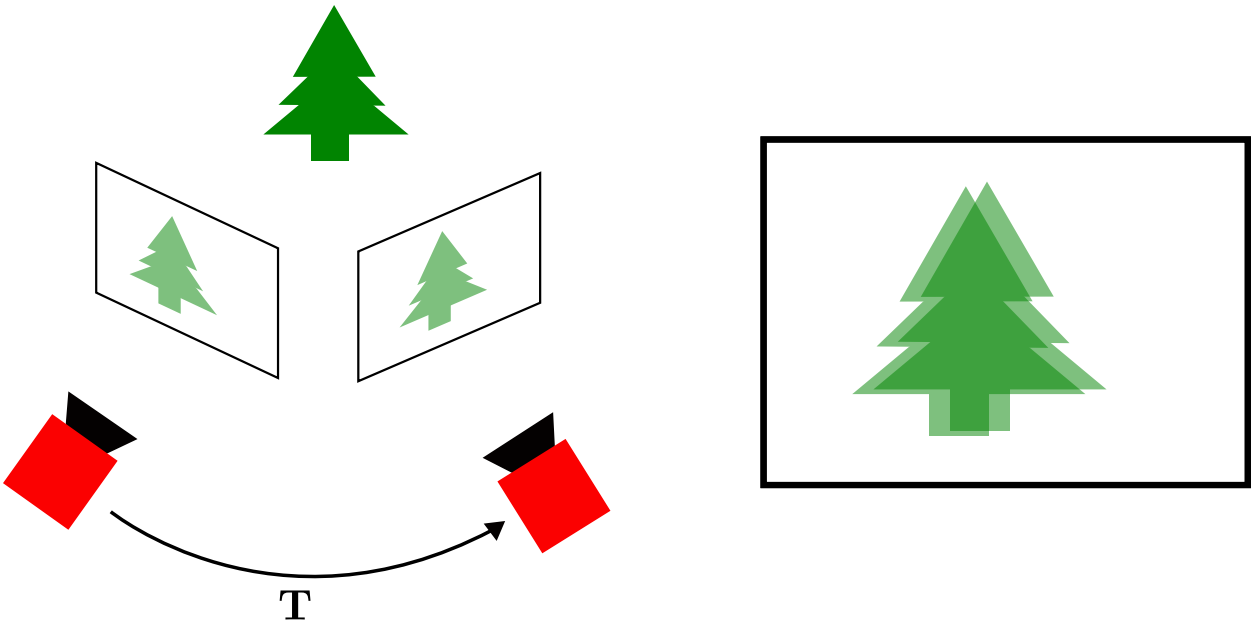



Figure 1.6: Illustration of projective association used by some registration techniques such as that of Kerl, Sturm, and Cremers (2013). Projective association works by transforming the data collected by one sensor into the reference frame of another. These approaches attempt to maximize photo-consistency.

bounded a point cloud in a sphere defined by a transformation cuboid to find the optimal point cloud registration. Parra Bustos, Chin, and Suter (2014) presented a similar approach, but used stereographic projections to bound the possible point locations. Izatt, Dai, and Tedrake (2017) formulate the point cloud registration problem as a mixed-integer program (MIP) by introducing a variable to represent what subregion of $\mathrm{SO}(3)$ they are currently evaluating in their branch and bound algorithm. Campbell et al. (2017) alternated between bounding map points on rotation and transformation for camera based localization.

The naive way of performing a search is exhaustively searching over all possibilities. Multiresolution search is related to branch and bound, but functions by exhaustively searching a lower resolution of the data that represents a bound on the full resolution of the data. Using the fact that after applying the rotation component of the $\mathrm{SE}(3)$ transformation, the translation component of that transformation is axis aligned reference frame of the sensor. For that reason, on gridded data, bounds can be efficiently computed (Olson (2009); Wolcott and Eustice (2017b)).

### 1.1.4 Application of Sensor Registration

Fundamentally, as stated earlier, sensor registration is the processes of estimating the rigid body transformation between two measurements and can be used in a variety of applications. This

can include consecutive measurements (producing an odometry estimate), measurements from disparate points in time (such as for localization and mapping), and between different sensors (such as for calibrating multiple sensors on a platform).

For odometry estimation, the transformations tend to be small and runtime is an important consideration. Consecutive measurements are aligned to produce a trajectory estimate which is locally accurate but can drift over longer distances since no loop closures are estimated. An example of such a method is that presented by Kerl, Sturm, and Cremers (2013). The timing considerations of odometry estimation is why they choose to use projective association, which is an approach that has drawbacks but is generally faster than other approaches.

In mapping applications, registration can be run offline, or at least at rates slower than real time. For these applications, the registration results provide transformation estimates between measurements. In factor graph or pose graph formulations of simultaneous localization and mapping (SLAM), these would be relative pose faction. Carlevaris-Bianco, Ushani, and Eustice (2015) used registration results as factors in a SLAM system to produce "ground truth" pose estimation results for their dataset.

Most approaches to calibration are done offline before the robotic system is used. In these applications, there are not usually time constraints and more thorough systems can be used. Wolcott and Eustice (2017a) calibrate a set of LIDAR sensors using registration results in a joint optimization framework.

## 1.2 Semantic Classification

Semantic classifiers are a set of predictive algorithms that label input data with a meaningful semantic class. For our work, the input data is a point cloud $\mathcal{X}$ along with per point RGB or intensity data, while the labels $\mathcal{S} \triangleq \{s_i\}$ are from a set of semantic classes $s_i \in \mathcal{C}$.

A combination of new tools to leverage the fine grain parallelism of a graphics processing unit (GPU) and the increasing size of labeled datasets has led to rapid advances in the speed and performance of semantic classification algorithms. Convolutional Neural Networks (CNN), such as the architecture shown in Figure 1.7, were enabled by the parallelism of GPUs to train a large number of parameters in a multilayer, nonlinear, decision function. These decision functions need a diverse set of training data to avoid over fitting and maximize utility. Examples of such datasets that have been used to successfully train CNNs are the general computer vision datasets ImageNet by Russakovsky et al. (2015) and MS COCO by Lin et al. (2014), as well as the mobile robotics specific KITTI semantics by Alhaija et al. (2018) and Cityscapes by Cordts et al. (2016). CNNs have done very well on semantic classification benchmarks (Krizhevsky, Sutskever, and Hinton (2012b); Shelhamer, Long, and Darrell (2017); He et al. (2017)).

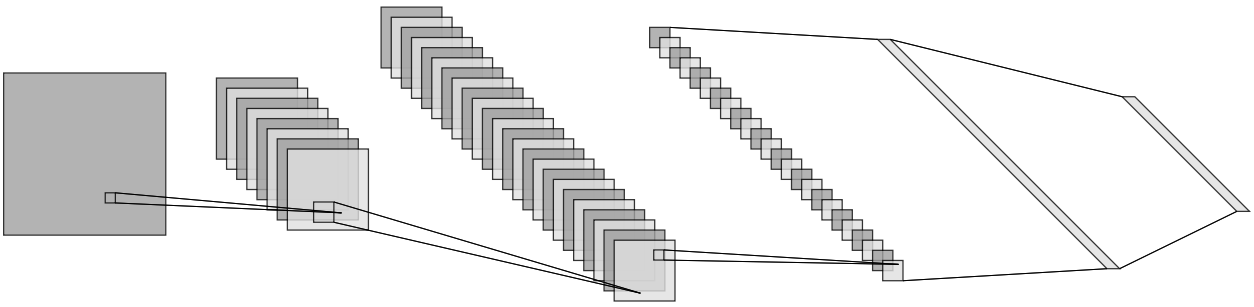


Figure 1.7: An example CNN architecture shows multiple layers of pooling and convolution.

Figure 1.8: Sample annotations from the MS COCO dataset. This dataset focused on evaluating algorithms that identify, label, segment, and caption everyday objects.

These datasets were developed to aid training in specific tasks. ImageNet was constructed to provide a large set of images with hierarchical labels of what the image depicts (e.g, an image could depict a mammal that is a cat). A subset of the images also provide a bounding box of the label it is depicting, allowing for the training and evaluation of object localization algorithms. MS COCO is a more resent and larger dataset that aims to aid a variety of object recognition and scene understanding tasks. An example of labeled data from the MS COCO dataset is shown in Figure 1.8. While they are focused on general computer vision tasks, these datasets can be used to aid indoor robotic systems.

KITTI semantics and Cityscapes are more specialized datasets. They focus on providing training and evaluation for perceptual systems of autonomous vehicles. Because autonomous vehicles typically have LIDAR and RGB-D sensors, these datasets are more likely to include 3D annotated data. They mostly focus on streets scenes, and include data from a variety of rural, suburban, and

urban environments. The labels in theses datasets focus on objects that are important to the operation of a vehicle, such as roadway, traffic signs, pedestrians, and other vehicles. Figure 1.9 gives an example of labeled data from the Cityscapes dataset.

Initial work on estimating semantically meaningful labels on images focused on labelling canonical photos with the class of object they depict (Krizhevsky, Sutskever, and Hinton (2012a)). It then was extended to classification and bounding box detection for objects in images (Girshick et al. (2014)), and then to pixel level semantic segmentation of photos (Long, Shelhamer, and Darrell (2015a). Pixel level semantic classification is what is used in Chapter 2 and Chapter 4 to aid registration. This work has led to advances in related fields. These include visual question answering tasks (Antol et al. (2015)) and captioning of images (Zhou et al. (2017)). Recently, there has been work toward panoptic segmentation (Kirillov et al. (2019)) which combines the tasks of pixel level semantic and instance segmentation.

## 1.3 Semantics and Geometry

Semantic classifiers can provide probabilistic labels of things in the environment $p(s_i|\mathcal{X})$. As we discussed earlier, combining semantics and geometry is an enabling technology for mobile robotics. It will enable them to perform tasks in unstructured, real-world, environments. Some examples of downstream tasks that would benefit from a joint semantic and geometric understanding are autonomous vehicle path planning and retrieval tasks for home robotics. Path planning would benefit from knowing what other types of road users are around the autonomous vehicle to better predict their behavior. Retrieval, by necessity, needs to jointly know where and what the object is that it is tasked with getting. In general, assistive home robotics will need a semantic understanding of the environment that matches the human it is assisting, and will need a joint semantic and geometric understanding to then interact with that environment.

In an attempt to combine semantics and geometry, dense 3D priors of objects have been used for scene estimation and mapping. Salas-Moreno et al. (2013) align 3D mesh model priors of objects to the RGBD frame. The technique treats objects as landmarks and each alignment as a factor in the SLAM framework. Choudhary et al. (2014) also use objects as landmarks, but instead of having a dense 3D prior over every object, the objects are discovered via segmentation and modeled during the mapping process. Bowman et al. (2017) approach data association in SLAM using EM, though at the sparse object level as opposed to the dense point level. There has been research on fusing a sequence of probabilistic label to create a semantic map including the work by McCormac et al. (2017a). Jadidi et al. (2017) use Gaussian process regression to construct a dense semantic map, as shown in Figure1.10. So far, these methods only overlay the semantic predictions on metric maps, and do not jointly optimize over both semantics and geometry.

Figure 1.9: Finely annotated data from Cityscapes dataset. The dataset includes 5000 such images. Large datasets such as this have enabled rapid advances in semantic classification.

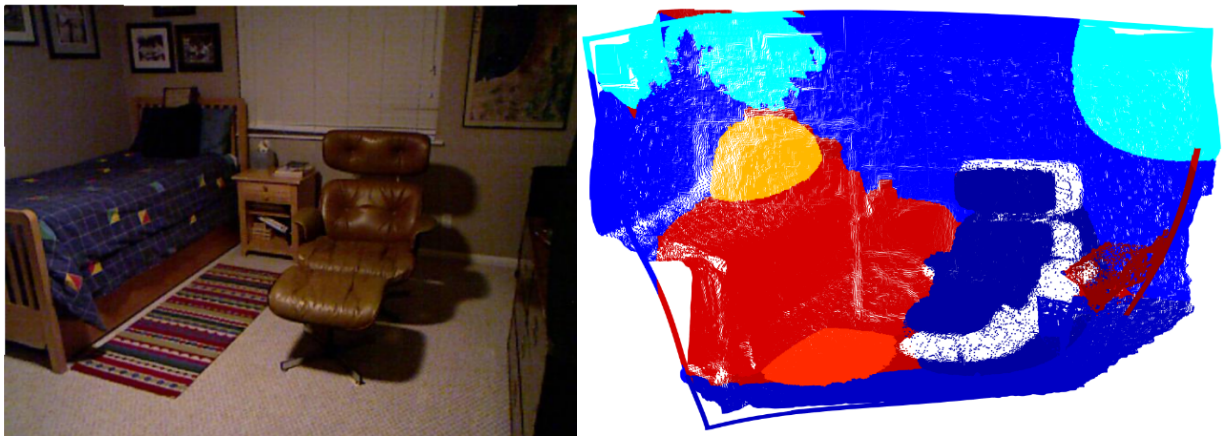


Figure 1.10: Dense semantic map produced using a Gaussian process classifier presented by Jadidi et al. (2017). Semantic mapping is one area that would benefit from improved accuracy and consistency in semantic sensor registration.

There has been work on robotic tasks aided by a joint semantic and geometric understanding of the environment. Sui et al. (2017) presented an approach where they sequentially estimated the geometric and semantic representation of the scene, and accounted for a robot manipulating an object in their scene hypothesis. Pronobis, Riccio, and Rao (2017) developed a system wherein a hierarchical semantic representation of a scene was used to predict affordances of a robot to aid in motion planning. Maturana, Arora, and Scherer (2017) presented a system where a micro-aerial vehicles (MAV) is used to build a semantic map of the environment to aid in scouting for objects of interest. Zeng et al. (2018) presented a mapping framework that combines RGB-D observations with contextual object information to more quickly and accurately detect object poses in cluttered scenes.

Most of these approaches are designed to create a 3D map of the environment after a trajectory has been estimated. This thesis is focused on the registration task, which is an element of the trajectory estimation system. Therefor, the work we present shows how to combine semantics and geometry to improve registration. These improvements to registration can then allow for more accurate maps, and therefore aid downstream robotic tasks.

Figure 1.11 shows why semantic classification estimation might be useful to registration, and vice versa. Semantic classifiers are trained to be consistent with their measurements, which provides valuable information to registration algorithm. And registration algorithm allows multiple measurements to accurately fused, for a more accurate estimate of what is in the environment. A joint understanding will not only enable robots to perform more task, it will also decrease the performance gap between humans and robots. This necessary to increase automation and to increase the coordination between robots and the humans that interact with them.

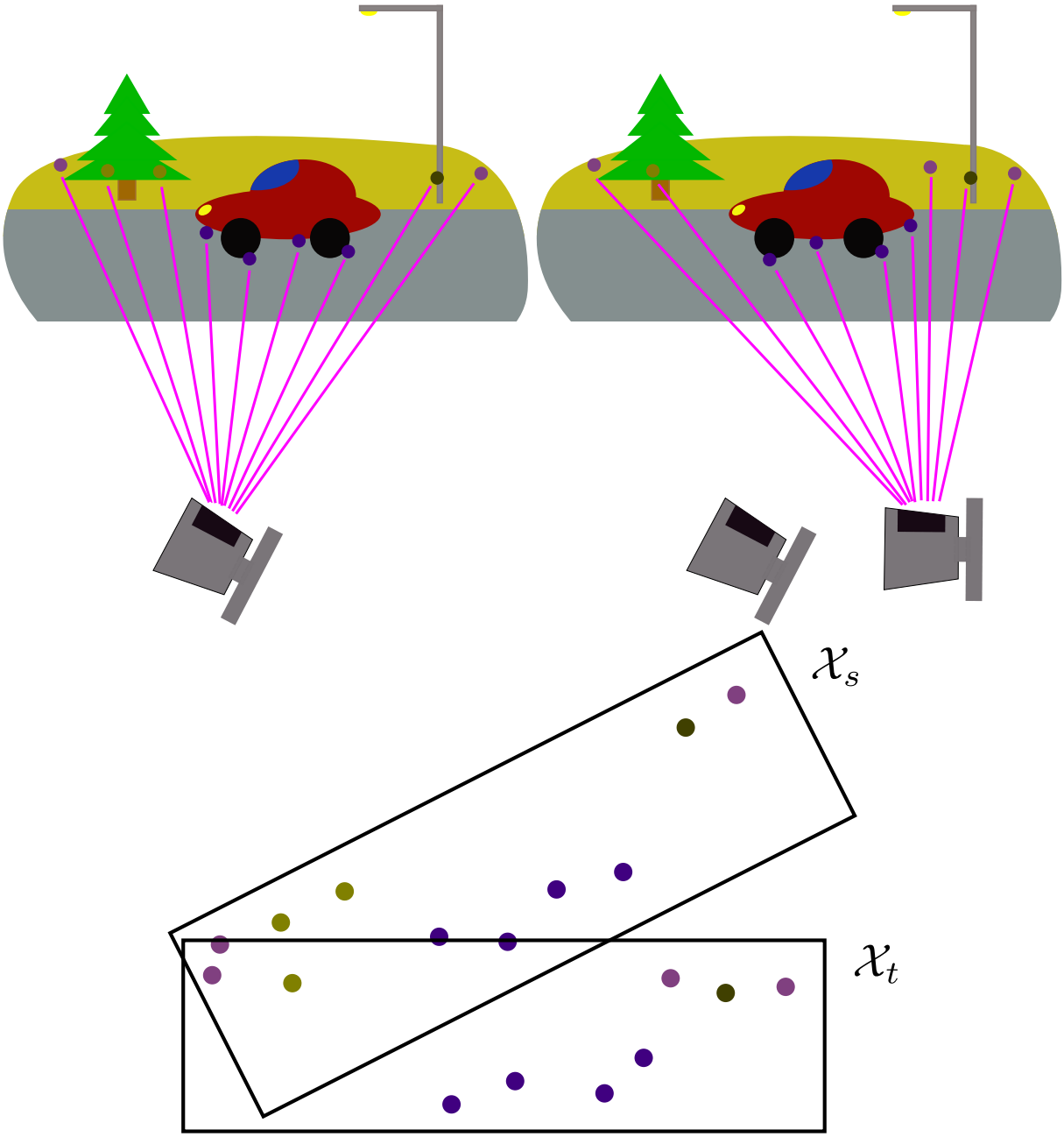



Figure 1.11: An illustration of the combination of 3D sensing and semantic classification

Advances in semantic classification have mostly come in lock-step with the release of larger datasets that contain more detailed labels. These improvements are also correlated with release of GPUs with greater and greater compute capabilities, along with libraries and resources to more effectively use and manage them. In the next section we will look more closely at some of the datasets that enabled these advances, and advances in the broader field of mobile robotics.

## 1.4 Datasets for Semantics and Geometry

In mobile robotics, trajectory and odometry estimation are well studied problems that are important for many tasks. In many cases, odometry provides the local state estimate used by planning and control systems. Due to this, there exist many publicly available datasets and benchmarks for odometry and trajectory estimation for mobile robotic systems.

Generally, these benchmarks and datasets can be split into three categories: autonomous driving, indoor scenes, and simulated. These categories vary in the sensor data available, and relatedly the availability and nature of "ground truth" results available for benchmarking.

Autonomous driving datasets, such as the KITTI dataset by Geiger et al. (2013) and the cityscapes dataset by Cordts et al. (2016), are collected on vehicle platforms and usually in street scenes commonly encountered while driving. They usually include LIDAR, mono camera, and stereo cameras for perception sensors. More recently, companies involved in autonomous vehicle development have open-sourced some of the data they use internally, including Waymo (2019) and Lyft (by Kesten et al. (2019)). There are some outdoor robotic datasets that include data from similar environments such as the NCLT dataset by Carlevaris-Bianco, Ushani, and Eustice (2015).

Indoor datasets have been driven by advances in artificial reality and virtual reality hardware, such as real-time stereo and structured light sensors. Examples of indoor datasets include the TUM RGB-D dataset by Sturm et al. (2012) and the NYU dataset by Nathan Silberman and Fergus (2012).

Simulated datasets can either simulate driving scenes, like the work by Johnson-Roberson et al. (2017), or indoor scenes, like the work by McCormac et al. (2017b). They are distinct because they offer truly ground truth results for both semantic classification results, and trajectory estimates. They also avoid the timely manual labelling process required for real world datasets, and therefore can proved more training data. The down side is that there might be idiosyncrasies of a particular simulation that prevent a method trained on it from generalizing to real world data. Essentially an algorithm trained on a system may learn to predict the physics engine used to generate the scene, and not the real-world environment that it is trying to simulate.

Some contemporary work in registration focus on a one of these categories, or even a subset of a category. In this thesis we aimed to design and study general purpose algorithms for registration.

Therefore, we evaluated our each of our contributions on datasets from several of these categories. We will describe more thoroughly three of the datasets we used in the work presented in this thesis.

### 1.4.1 KITTI Dataset

The KITTI dataset by Geiger et al. (2013) was one of the first publicly available autonomous vehicle dataset. It was first introduced in 2013 but they have continued to release new benchmarks and labeled training data. It is a set of data collection sequences and benchmarks to evaluate various aspects of an autonomous vehicle system includeing odometry, object detection, stereo matching, and scene flow. In this work we mostly focused on the odometry benchmark but in Chapter 2 we also use the available semantic label data for training. The odometry dataset is a series of 21 sequences of street scenes collected in Germany in various urban and rural environments. Evaluation of trajectory estimation accuracy available via comparison to a fused global positioning system (GPS) and inertial navigation system (INS) system. The benchmark is computed as transformation errors for all possible subsequences of (100, 200, ..., 800) meters. This evaluation data is available for the first 11 sequences for training and tuning of proposed algorithms. For the next 5 the evaluation data is not available but results can be uploaded for a detailed evaluation of their performance. The final 5 sequences can be uploaded but only summary results are published. The available sensor data include stereo camera data and a 64 beam spinning 3D LIDAR sensor.

Figure 1.12 shows results from sequence 00 of the odometry benchmark where odometry is computed using the GICP algorithm between sequential point clouds.

### 1.4.2 TUM RGB-D Dataset

The TUM RGB-D dataset by Sturm et al. (2012) is an indoor dataset collected with the Microsoft Kinect sensor. The Kinect is a structured light sensor that projects an infrared pattern that it uses to measure the depth of the scene. It also has an inertial measurement unit (IMU) sensor whose data is included in the dataset. The dataset includes a variety of sequences with various structures and textures to evaluate the performance of RGB-D SLAM and odometry sequences in a variety of environments. It also includes sequences with large loop to test systems with loop-closure capabilities. Most of the data was collected with a handheld Kinect sensor, but some of the sequences were collected by mounting the Kinect on a indoor mobile robot platform and manually navigating it through the scene. Evaluations are available with respect to the trajectory provided by a high speed motion capture system. The benchmark includes a relative metric for odometry estimates and a absolute metric for SLAM algorithms. Figure 1.13 shows sample data from the dataset from a sequence collected around a stereotypical scene of a messy desk belonging to a researcher.

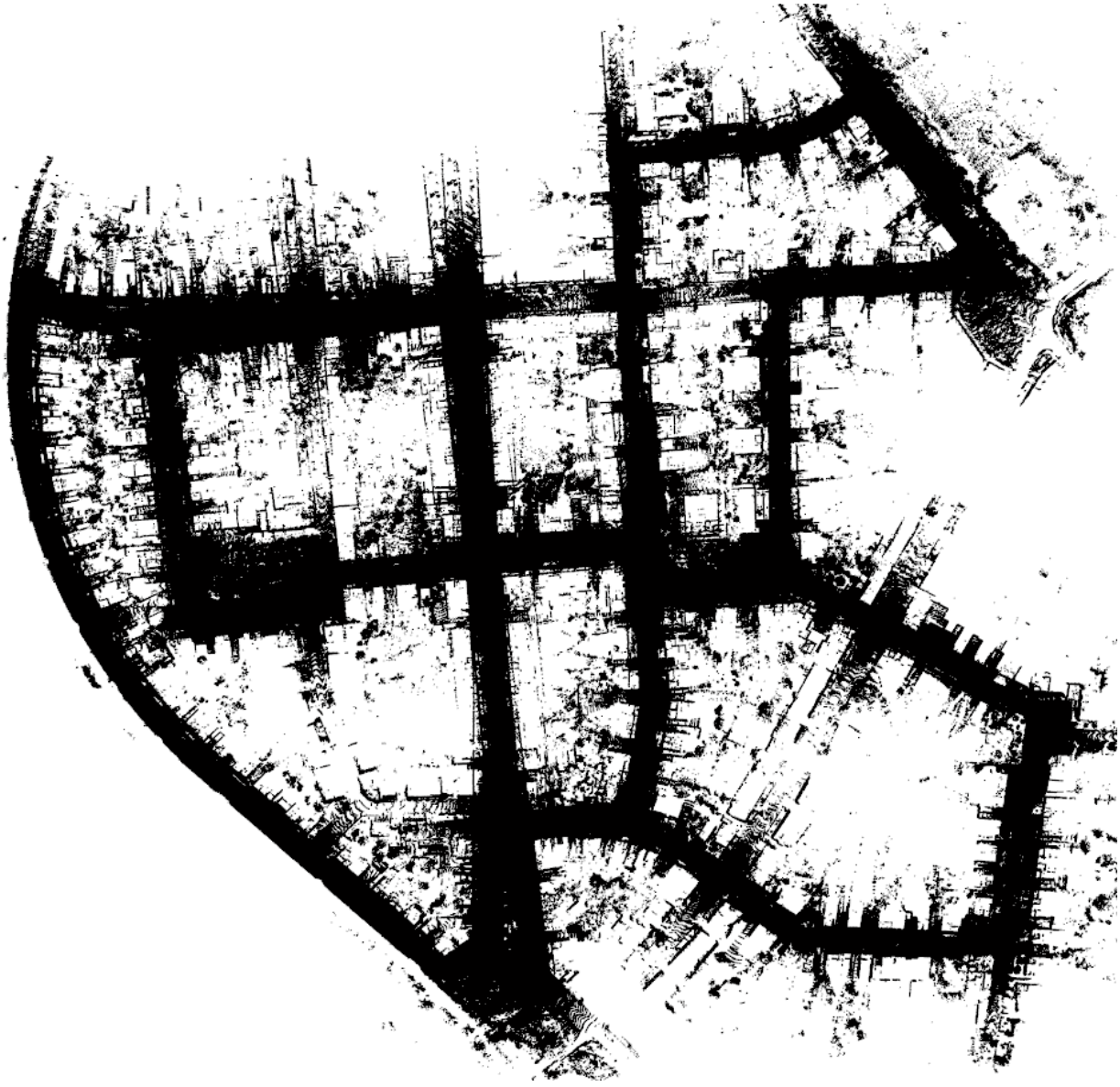


Figure 1.12: An example of an trajectory sequence of the odometry benchmark of the KITTI dataset. Sequential LIDAR scans were registered using the GICP algorithm and transformed into the same frame of reference using the estimated trajectory.

Figure 1.13: Sample data from the TUM RGB-D dataset. The dataset includes a variety of indoor scenes, such as the cluttered desk scene shown in this picture.

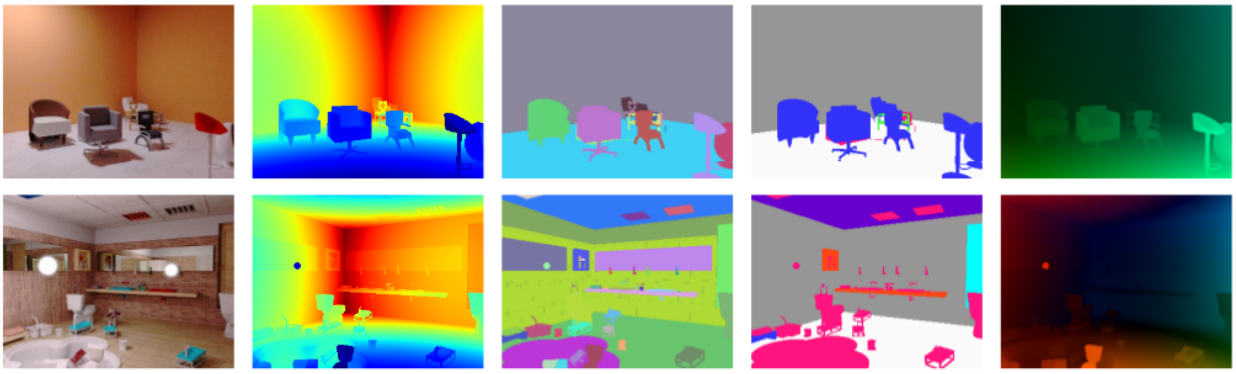


Figure 1.14: An example provided by McCormac et al. of the ground truth data available with the SceneNet RGB-D dataset. From left to right: photorealistic image, depth, object instance segmentation, semantic class segmentation, optical flow. Photo courtesy of McCormac et al.

### 1.4.3 SceneNet RGB-D Dataset

The SceneNet RGB-D Dataset by McCormac et al. (2017b) is a synthetic dataset of 15,000 randomly generated indoor scenes that aims to have photorealistic quality. Most previous indoor datasets, such as the NYU dataset (Nathan Silberman and Fergus (2012)) rely on a small amount of hand labeled data. In contrast, SceneNet RGB-D can randomly generate a large number of trajectories for training machine learning algorithms to supplement real world datasets. Being synthetic allows for a variety of ground truth information, such as what is pictured in Figure 1.14, including depth, instance segmentation, class segmentation, optical flow. They use a physics engine to generate valid 3D rooms populated with object models from ShapeNet (Chang et al. (2015)), randomly generated wall textures, and ceiling lighting. They then randomly sample trajectories through the scene, weighting the distribution of potential positions to more likely choose those that point towards the center of the room. While useful in many ways, the dataset has drawbacks which include physically valid but unpractical room layouts (e.g., a bathroom with stacked toilets, a living room with the seating facing the wall) and the scenes are static besides the movement of the camera.

## 1.5 Thesis Outline

The work proposed in this document seeks to extend the state-of-the-art in joint semantic, geometric sensor registration. Since many of the algorithms were first proposed 20 to 30 years ago, advances in hardware capabilities and algorithmic methods encouraged a revaluation of approaches to sensor registration. In addition, new methods in semantic classification have encouraged new application for mobile robotics. We hoped to find new ways to combine these fields, to help enable new applications of mobile robotics. Particularly, there is the potential for new hardware and

improved performance of semantic classifiers to aid the accuracy of registration by improving the accuracy and robustness of the data association between target and source data. Towards this, we propose the following contributions of this thesis:

- We present an algorithm for joint semantic-geometric point cloud registration that uses the expectation-maximization (EM) approach. In this approach semantic classification results from a CNN inform the data association in the registration problem. Not only does this approach minimize geometric residuals, it also maximizing semantic consistency, increasing the reliability of down stream tasks joint semantic-geometric tasks. This work is discussed in Chapter 2.

  **S. A. Parkison**, L. Gan, M Ghaffari, and R. M. Eustice. *Semantic Iterative Closest Point through Expectation Maximization*. In Proceedings of the British Machine Vision Conference, Newcastle, UK, September 2018.

  **Repository:** `https://bitbucket.org/saparkison/perl_registration`

  This repository includes a c++ implementation of the method proposed in Chapter 2. We also include code to evaluate the results on the KITTI odometry dataset, and a visualization tool to view the results. In addition, this repository includes an implementation of the KD tree data structure on the GPU, a GPU GICP algorithm, and a version of the proposed method that runs on the GPU.

- We present work to refine sensor registration by using viewpoint invariant features. The method in Chapter 2 is a local, gradient-based method similar to other ICP algorithms. We utilize viewpoint invariant features, namely intensity, for better alignments and convergence. We do this by training a sparse Bayesian representation of the invariant feature and use the distance between regressed representation as a regularizer on the registration problem. This work is discussed in Chapter 3.

  **S. A. Parkison**, M Ghaffari, L. Gan, R. Zhang, A. K. Ushani, and R. M. Eustice. *Boosting Shape Registration Algorithms via Reproducing Kernel Hilbert Space Regularizers*. In IEEE Robotics and Automation Letters, Volume: 4 , Issue: 4 , October 2019.

  **Repository:** `https://bitbucket.org/saparkison/rkhs_gicp`

  This repository includes an c++ implementation of the method proposed in Chapter 3 and code to evaluate the proposed method on the KITTI odometry dataset and TUM RGB-D dataset. This includes our c++ version of the sequential training method proposed by Tipping, Faul et al. (2003) (originally released as Matlab code) and our implementation of the multi-channel GICP algorithm proposed by Servos and Waslander (2017) (not released by the author).

- We present work to further improve sensor registration by reformulating it as a MIP which jointly solves for the data association and the rigid body transformation. This approach allows us to solve the registration problem without strong priors on the transformation or data associations. We do this for a two-dimensional (2D) to 3D registration problem, which can represent localizing a camera into a 3D map, or determining the extrinsic calibration between a camera and LIDAR sensor. We show this working with linear semantic features such as poles, curbs, and road paint. This proposed research is discussed in Chapter 4 and under review for ICRA 2020.

  **S. A. Parkison**, J. M. Walls, R. W. Wolcott, M. Saad, and R. M. Eustice. *2D to 3D Line-Based Registration with Unknown Associations via Mixed-Integer Programming*. In IEEE Conference on Robotics and Automation, 2020 (*to appear*).

  **Repository:** `https://bitbucket.org/saparkison/plucker_line_mip`

  This repository includes our Julia implementation of the method proposed in Chapter 4. Or approach uses the JuMP library that provides an abstraction to interface off the Gurobi Optimization library, but which can be used with several other open source MIP solvers. It also includes our implementation of the singular value decomposition (SVD) approach proposed by Přibyl, Zemčík, and Čadík (2016) (originally released as Matlab code) and evaluation code for the VGG dataset. This repository additionally includes our proposed approach to extracting linear semantic features from images.

### 1.5.1 Document Roadmap

This thesis aims to present these contributions and evaluations of the proposed approaches on some of the publicly available datasets presented in Section 1.4 with comparisons to state of the art registration algorithms. The next three chapters aim to more thoroughly present these algorithm and explain their relationship to other methods used in the robotics community. The final chapter summarizes the contributions and presents interesting future work that was not addressed by these contributions. We also provide several appendices for adjacent topics that are relevant to understanding our contributions. The rest of this document is arranged as follows:

- **Chapter 2**

  We present our approach to semantic sensor registration using the expectation maximization technique published in (Parkison et al., 2018). We derive our expectation maximization formulation where semantic classification probabilities to inform the association in the registration formulation. The second step of minimizing the residual becomes a weighted least squares. We present evaluation on the KITTI and SceneNet RGB-D datasets comparing to

other state of the art algorithms on accuracy and convergence properties.

- **Chapter 3**
  We present our work on using sparse Bayesian approach to regress function to represent intensity information to use a regularizer for registration problems published in (Parkison et al., 2019). Our method avoids a reliance on data associations between target and source by minimizing the distance between two regressed functions. We compare our method to state of the art methods using the KITTI odometry (using LIDAR) data and TUM RGB-D datasets. We compare to state of the art method including other methods that use extra channels multi-channel GICP (Servos and Waslander (2017)) and GICP 6D (Korn, Holzkothen, and Pauli (2014)) on relative accuracy and convergence.

- **Chapter 4**
  We present our work on reformulating the 2D to 3D sensor registration as a MIP and published in (**?**). We do this for Plücker line coordinates. We show how we linearized the cost function and our approach to treating the association variable as a binary variable. We evaluate are results on the VGG multiview dataset and data collected on a autonomous vehicle platform using linear semantic features. We compare to an SVD based approach presented by Přibyl, Zemčík, and Čadík (2016).

- **Chapter 5**
  Summarizes the contributions presented in this thesis and presents future work. This future work includes parallelizing the contributes on to a GPU for increased runtime efficiency, estimation in dynamic scenes that may be represented by several rigid body transformation, eliminating data association from the registration problem, and creating bespoke optimization algorithms to leverage domain specific knowledge.

- **Appendix A**
  Gives an introduction to Lie Groups, specifically $\mathrm{SO}(3)$ and $\mathrm{SE}(3)$. These groups represent 3D rotation and transformation. $\mathrm{SE}(3)$ represents the parameter we are trying to estimate in many of these registration tasks.

- **Appendix B**
  Presents the optimization over manifold (such as what is presented in Appendix A) techniques used in Chapter 2 and Chapter 3. These approaches let us use optimization algorithms designed for Euclidean spaces on manifolds, which are smooth groups that are locally Euclidean at each point. The Lie Groups presented in Appendix A are examples of such manifolds.

- **Appendix C**

  Gives an introduction of Mixed Integer Programming used in Chapter 4. MIP seeks to solve an optimization problem where some of the variables are integer valued. MIP are NP-complete, but there are many off the shelf solvers that use heuristics to increase the likelihood that a solution is found quickly in most practical problems. The subclass of MIP we are interested is mixed integer linear programs, where the cost function and constraints are linear, except for the integer constraint on some of the variables.

- **Appendix D**

  Provides an overview of the software the implements the contributions of this thesis. This section provides an overview of that software and explains how to use it and any evaluation software provided.

# CHAPTER 2

# Semantic Iterative Closest Point Through Expectation Maximization

In this chapter we present a method for sensor registration that jointly minimizes geometric residuals and maximizes semantic consistency between measurements. Many current approaches combining geometry and semantics treat them as problems to handle separately. In this work, we present an approach based on Expectation Maximization where the association between measurements is treated as a latent variable, and is conditioned on the probabilistic semantic labels provided by a Convolutional Neural Networks (CNN) based classifier. Not only does this provide more accurate transformation estimates, but it has the possibility to improve downstream task such as semantic mapping or object tracking. We present results on two publicly available datasets, the KITTI Odometry dataset and the SceneNet RGB-D dataset and show our method more consistently coverages to the correct transformation.

## 2.1 Introduction

Point cloud registration, the task of finding the rigid body transformation between two point clouds, is an integral part of geometric inference in many modern perception systems. The most successful algorithm is known as the Iterative Closest Point (ICP) aglorithm byBesl and McKay (1992) and extended by Chen and Medioni (1991a). ICP was further developed to the probabilistic framework known as Generalized ICP (GICP) by Segal, Haehnel, and Thrun (2009).

Semantic inference on images and point clouds has shown increasing value in vision-based applications. Early algorithms relied on classifiers trained on a set of hand-crafted features (Brostow et al. (2008); Shotton, Johnson, and Cipolla (2008)). However, their computational efficiency limits their application in real-time scenarios. Advances in Convolutional Neural Networks (CNNs) have improved the computational efficiency of semantic segmentation techniques with superior performance in both indoor and outdoor benchmarks (Long, Shelhamer, and Darrell (2015b); Yu

and Koltun (2016); He et al. (2016); Qi et al. (2016); Qi et al. (2017)). Together with pose estimation techniques, multiple scenes can be segmented and combined to perform semantic mapping (McCormac et al. (2017a)); nevertheless, most semantic mapping research has focused on combining geometry and semantics into a map representation, and not on how semantics can improve pose estimation.

In this chapter, we develop the Semantic ICP algorithm that directly incorporates pixel semantics into the registration problem between two overlapping point clouds. The primary motivation is aiding tasks that rely on joint semantic segmentation and relative pose estimation, such as semantic mapping and object tracking. Figure 2.1 illustrates this concept where semantic labels in an indoor scene aid the alignment. In particular, this work has the following contributions:

1. Development of the Semantic ICP algorithm which uses joint semantic and geometric probabilities for finding the associations in the GICP-$\mathrm{SE}(3)$ algorithm, where GICP-$\mathrm{SE}(3)$ algorithm solves the point cloud registration problem with respect to the motion group manifold structure.

2. The open source implementation of the proposed algorithms as well as code to reproduce the provided results [1].

3. We provide experimental evaluations using publicly available benchmarks, KITTI (Geiger, Lenz, and Urtasun (2012)) and SceneNet RGBD (McCormac et al. (2016, 2017b)) datasets, that show improved registration performance over current methods.

## 2.2 Related Work

Point cloud registration is generally formulated as an optimization problem over the rigid body transformation that minimizes some residual between points in the source cloud to points in the target cloud. The ICP algorithm by Besl and McKay (1992) defines the objective function as the Euclidean distance between points in the source cloud, to an associated point in the target cloud. That association is rarely known and is unobserved by the sensor, thus the approach taken by Besl and McKay (1992) is to alternate between finding the Euclidean Nearest Neighbor (NN) association between points in each cloud and optimizing the point-to-point distance function over the transformation variables. This approach is generalized slightly using point-to-line by Censi (2008) and point-to-plane by Chen and Medioni (1991a) objective functions that have been shown to improve convergence speed and accuracy, by both Rusinkiewicz and Levoy (2001) and Pomerleau et al. (2013).

[1] https://bitbucket.org/saparkison/perl_registration

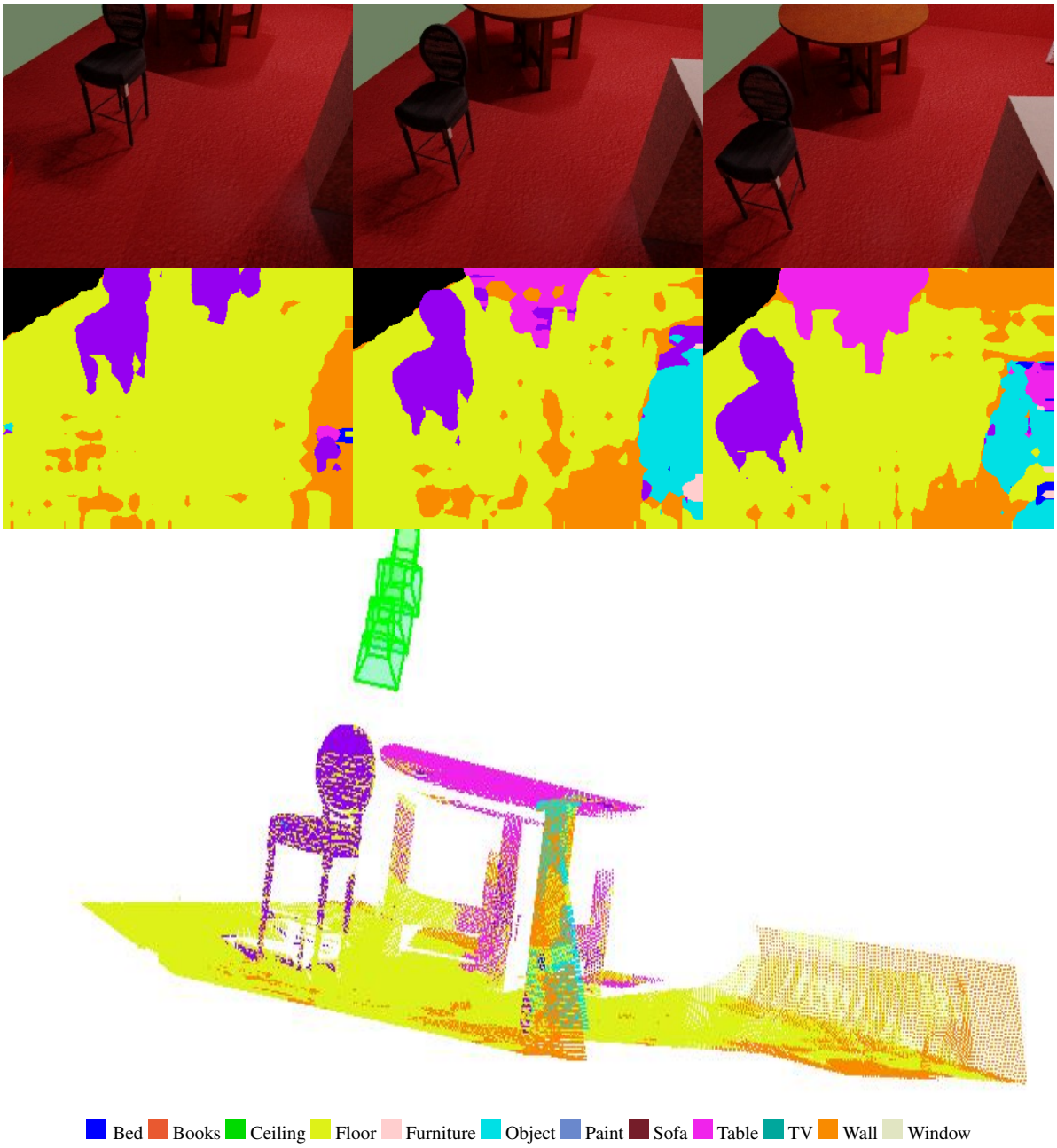



Figure 2.1: An example of three point clouds from the SceneNet RGBD dataset aligned using Semantic ICP. The left column shows the source images. The middle column shows the inferred semantic classes, labeled by the most likely class. The right figure shows three point clouds projected into a common reference frame by the estimated transformations, with the camera positions marked in green. Class labels are indicated below the image. The crisp objects are a sign of good alignment. Moreover, data association is addressed through performing a joint semantic and geometric inference by using the EM technique in which semantic labels and point associations between two point clouds are treated as latent random variables. Despite inaccuracies in the inference, semantics improve point cloud registration results.

There has also been work on defining probability distributions over points in the point clouds. Biber and Straßer (2003) define normal distributions using points in the target point cloud that fall into voxels of the environment. The objective function is defined as the probability that a point in the source point cloud is within the distributions of the target point cloud. Generalized ICP by Segal, Haehnel, and Thrun (2009) also defines a Gaussian distribution over the source and target point clouds, but computes these distributions by calculating the sample covariance of neighboring points, where neighbors are those points that are the closest in the Euclidean distance. These probabilistic formulations of ICP and the iterative nature of the algorithm have led to expectation-maximization (EM) approaches to the point cloud registration problem, including that by Granger, Pennec, and Roche (2001). Lee and Lee (2016) use an EM approach to align sensor measurements to 3D models of objects while also learning the covariance of the observation to improve the alignment. Their method performs well on the model alignment task, but they treat the distribution on model and observation points differently, which does not generalize to point cloud registration between two observations. Gabriel Agamennoni and Sorrenti (2016) also formulate point cloud registration as an EM problem. They model points in the source cloud drawn from a t-distribution centered at points in the target cloud. None of these methods include semantics. Our approach, which includes semantics, improves upon these methods' registration accuracy, as we show in the evaluation.

We conduct joint geometric and semantic inference to improve the registration task. Semantics have been combined with geometry in a variety of ways. Object level classification has been used on premade maps (Castle et al. (2007); Civera et al. (2011)). Bao and Savarese (2011) use an object detector in the structure-from-motion setup to jointly estimate camera parameters, 3D points, and object instances and poses. Their method, unlike ours, requires parametrizing objects, and only estimates sparse object poses and not dense point labels. These approaches are developed to improve scene estimation by providing more geometric constraints. Conversely, Pillai and Leonard (2015) use monocular SLAM to aggregate multiple views of a single object to provide more evidence to the object detector. This approach treats pose estimation and semantic classification as independent, solving the first to improve the second.

Dense 3D priors of objects have also been used for scene estimation and mapping. Salas-Moreno et al. (2013) align 3D mesh model priors of objects to the RGBD frame. The technique treats objects as landmarks and each alignment as a factor in the graphical SLAM framework. Choudhary et al. (2014) also use objects as landmarks, but instead of having a dense 3D prior over every object, the objects are discovered via segmentation and modeled during the mapping process. Bowman et al. (2017) approach data association in SLAM using EM, though at the sparse object level as opposed to the dense point level. The iterative nature of ICP is closer to that of an EM framework then SLAM is, which requires finding the solution to the full SLAM problem multiple

times at each step to converge on an association. Yu, Xiao, and Funkhouser (2015) use semantics in city-wide mapping by extracting semantic features from point clouds and then matching and aligning the features. This contrasts with our approach in that we propose to densely align points through joint geometric and semantic inference.

Sevilmis and Kimia (2016) leveraged shapes of objects to improve optical flow matching using richer representations such as scale-invariant feature transform (SIFT) and CNN features. It was found that the greater the visual variation between the images, the more their approach was aided by shape correspondences. There has been other work, which while it does not directly use semantic class labels, that uses object and feature geometry to improve association in the registration problem. Gressin et al. (2013) use feature computed on the local geometry around a point to both select good points to use and to improve association search. Similarly, Weinmann and Jutzi (2015) use the local geometry of a point to assess the quality, which improves the number of inliers for their RANSAC based registration method, while also improving the convergence rate. Zaganidis et al. (2017) propose an approach that adds semantics to the Normal Distribution Transform. In the latter work, their definition of semantics is geometric edges and planes. Instead, our definition is object and class labels. These last approaches also differ in that they strictly enforce NNs; whereas we treat semantics as noisy measurements that assist in modeling the probability of association.

## 2.3 Problem Statement and Formulation

We wish to find the 3D rigid body transformation that aligns two semantically labeled point clouds. We will be using $\mathcal{X} \subset \mathbb{R}^3$ to represent a set of spatial coordinates collected by a range/depth sensor. The following definitions are used throughout the chapter.

**Definition 1** (Target point cloud)**.** *The point cloud $\mathcal{X}_t$ which is considered to be in a fixed reference frame is called the target point cloud.*

**Definition 2** (Source point cloud)**.** *The point cloud $\mathcal{X}_s$ which $\mathbf{T} \in \mathrm{SE}(3)$ acts on is called the source point cloud.*

The action of $\mathbf{T}$ on any point $\mathbf{x}_i \in \mathcal{X}$ is $\mathbf{T}(\mathbf{x}_i) = \mathbf{R}\mathbf{x}_i + \mathbf{p}$, where $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{p} \in \mathbb{R}^3$. The likelihood function for aligning two point clouds sampled from the same environment depends on data association between them. We define the association variable $\mathcal{I} \triangleq \{i_k, j_k\}_{k=1}^n \in \mathbb{I}$ where $i_k, j_k$ indicate $\mathbf{x}_k^t \triangleq \mathbf{x}_{i_k}^t \in \mathcal{X}_t$ is a measurement of the same point as $\mathbf{x}_k^s \triangleq \mathbf{x}_{j_k}^s \in \mathcal{X}_s$, and $\mathbb{I}$ is the set of all possible associations (permutations). In short, the association set $\mathcal{I}$ gives the indices of points in the target and source cloud which are independent measurements of the same point. We also introduce a new random variable, $\mathcal{R} \triangleq \{\mathbf{r}_k\}_{k=1}^n$, to represent the residual where $\mathbf{r}_k \triangleq \mathbf{x}_k^t - \mathbf{T}(\mathbf{x}_k^s)$. To emphasize that the likelihood term includes the action of $\mathbf{T} \in \mathrm{SE}(3)$ on $\mathcal{X}_s$, we shall write the

log-likelihood function as $f(\mathbf{T}; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) \triangleq \log p(\mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}; \mathbf{T})$. However, for simplicity, we use $p(\mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I})$ whenever $\mathbf{T}$ is irrelevant.

Most ICP-based approaches follow an iterative two-step procedure for solving the point cloud registration problem. **Step 1**: Determine the association $\mathcal{I}$. **Step 2**: Minimize the cost defined using the residual, $\mathcal{R}$, over the parameter $\mathbf{T}$.

Thus, the geometric point cloud registration problem in **Step 2**, give a fixed set of associations $\mathcal{I}$, is expressed as follows.

**Problem 1** (Point cloud registration)**.** *Let $\mathcal{X}_t$ and $\mathcal{X}_s$ be two geometric point clouds. Given correspondences between target and source point clouds, $\mathcal{I}$, the optimal transformation that aligns the source to the target can be computed by solving the following maximum likelihood estimation (MLE) problem:*

$$\underset{\mathbf{T}\in\mathrm{SE}(3)}{\text{maximize}} \; f(\mathbf{T}; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) \tag{2.1}$$

Point clouds $\mathcal{X}_t$ and $\mathcal{X}_s$ observe the geometry of the environment; however, through the inclusion of semantic knowledge more information can be inferred. Let $\mathcal{C}$ be the set of semantic class labels. Define $\mathcal{S} \triangleq \{s_k\}_{k=1}^n$, where $s_k \in \mathcal{C}$. $\mathcal{S}$ represents the semantic class labels of points in the environment. Now $\mathcal{I}$ also encodes the association of a pair of points, one in $\mathcal{X}_s$ and one in $\mathcal{X}_t$, to a semantic label $s_k \in \mathcal{S}$. The joint distribution of the residuals $\mathcal{R}$, semantics $\mathcal{S}$, and association $\mathcal{I}$, conditioned on the source and target point clouds is $f(\mathbf{T}; \mathcal{R}, \mathcal{S}, \mathcal{I}|\mathcal{X}_t, \mathcal{X}_s) \triangleq \log p(\mathcal{R}, \mathcal{S}, \mathcal{I}|\mathcal{X}_t, \mathcal{X}_s)$. Thus, if the assumption of know associations is removed, the semantic point cloud registration is an optimization over the log-likelihood as follows.

**Problem 2** (Semantic point cloud registration)**.** *Let $\mathcal{X}_t$ and $\mathcal{X}_s$ be two independent, overlapping point clouds. Let $\mathcal{S}$ be the semantic labels of the environment observed by the point clouds. The optimal transformation that aligns the source to the target can be computed by solving the following MLE problem:*

$$\underset{\mathbf{T}\in\mathrm{SE}(3)}{\text{maximize}} \; f(\mathbf{T}; \mathcal{R}, \mathcal{S}, \mathcal{I}|\mathcal{X}_t, \mathcal{X}_s) \tag{2.2}$$

## 2.4 Generalized ICP on $\mathrm{SE}(3)$

Segal, Haehnel, and Thrun (2009) modeled measurements in the target and source clouds as being drawn from Gaussian distributions, i.e., $\mathbf{x}_k^t \sim \mathcal{N}(\hat{\mathbf{x}}_k^t, \mathbf{\Sigma}_k^t)$, and $\mathbf{x}_k^s \sim \mathcal{N}(\mathbf{T}(\hat{\mathbf{x}}_k^s), \mathbf{\Sigma}_k^s)$, respectively. Therefore, the residual log-likelihood, excluding the normalization constant, becomes $f_{\texttt{GICP}}(\mathbf{T}; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) \triangleq \sum_{k=1}^n ||\mathbf{x}_k^t - \mathbf{T}(\mathbf{x}_k^s)||_{\mathbf{C}_k}^2$ where $\mathbf{C}_k \triangleq \mathbf{\Sigma}_k^t + \mathbf{R}\mathbf{\Sigma}_k^s\mathbf{R}^\top$. The analytical gradient of this cost function in the ambient Euclidean space is $\frac{\partial f_{\texttt{GICP}}}{\partial \mathbf{p}} = \sum_{k=1}^n -2\mathbf{C}_k^{-1}\mathbf{r}_k$ with respect to the translation and $\frac{\partial f_{\texttt{GICP}}}{\partial \mathbf{R}} = \sum_{k=1}^n -2\mathbf{C}_k^{-1}\mathbf{r}_k(\mathbf{x}_k^{s\top} + \mathbf{r}_k^\top\mathbf{C}_k^{-1}\mathbf{R}\mathbf{\Sigma}_k^s)$ with respect to rotation. This

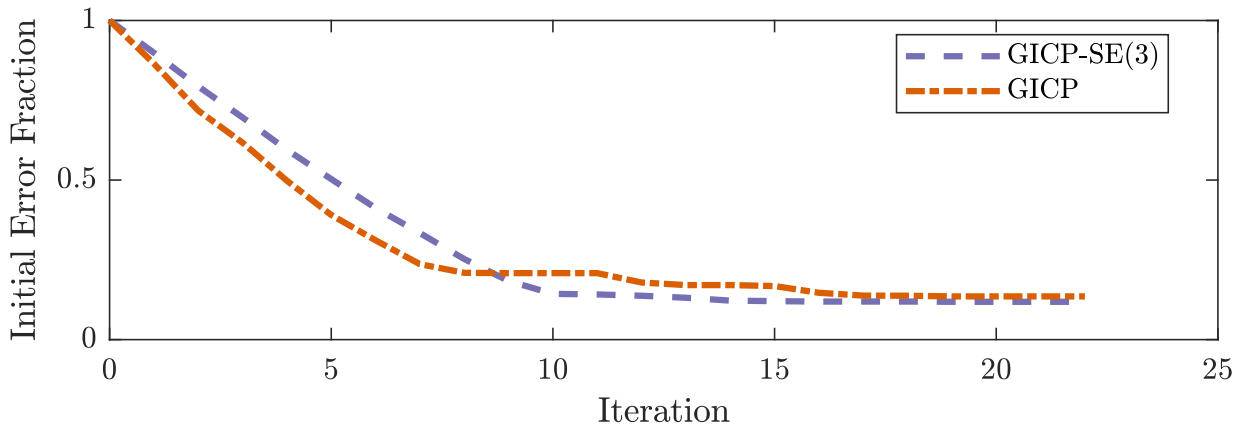



Figure 2.2: Convergence evaluation for GICP and GICP-$\mathrm{SE}(3)$. Median Fraction (taken from 50 alignments) of initial error, measure as $\mathrm{d}_{\mathrm{SE}(3)}(\cdot,\cdot)$, over outer loop iterations. While GICP initially converges faster, GICP-$\mathrm{SE}(3)$ reaches steady state in fewer iterations.

sets up Problem 1 as an optimization over the $\mathrm{SE}(3)$ manifold. While it does not change the formulation, optimizing over $\mathrm{SE}(3)$ is a more efficient way to parametrize the optimization problem with respect to the nature of the rigid body transformation, and by itself shows improvements over the Euler angle parametrization used in the original implementation of GICP (Segal, 2009). Our approach follows that of Absil, Mahony, and Sepulchre (2009), i.e., lifting the problem on to the tangent space of the Lie group, solving the reparametrized problem, then retracting it back to the manifold.

The original GICP algorithm removes residuals larger than a certain value to ensure that any point in the source cloud which does not have a counterpart will not affect the solution. To avoid having a hard threshold, we replace this step with a robust estimator using the Cauchy loss function, $\rho_\alpha(x) = \alpha^2 \ln(1 + \frac{x}{\alpha^2})$, where $\alpha$ is a parameter that controls where the loss begins to scale sublinearly. Similar to the approach in GICP, the robust estimator diminishes the effect of outliers while avoiding removal of potential inliers. Consequently, our cost function becomes

$$f_{\texttt{GICP}}(\mathbf{T}; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) = \sum_k^n \rho_\alpha(||\mathbf{x}_k^t - \mathbf{T}(\mathbf{x}_k^s)||^2_{\mathbf{C}_k}) \tag{2.3}$$

and the effect of the loss function on the gradient is trivial to derive using the chain rule. The algorithmic implementation of GICP-$\mathrm{SE}(3)$ is shown in Algorithm 1. For **Step 1** in the ICP framework, finding the association is done using a NN search in line 9. In **Step 2** (2.3) is solved by the lift-solve-retract scheme over the $\mathrm{SE}(3)$ Lie group as described in Appendix A. The inner loops are stopped once the change in $\mathbf{T}^*$ is less than a distance threshold $\epsilon$. We use a distance metric on $\mathrm{SE}(3)$ defined as $\mathrm{d}_{\mathrm{SE}(3)}(\mathbf{T}_1, \mathbf{T}_2) \triangleq \|\log(\mathbf{T}_1\mathbf{T}_2^{-1})^\vee\|$ where $\log(\cdot)$ computes matrix logarithm.

**Algorithm 1** GICP-$\mathrm{SE}(3)$

**Require:** Initial transformation $\mathbf{T}^{\text{init}}$, target point cloud $\mathcal{X}_t$, source point cloud $\mathcal{X}_s$;

$\mathbf{T}^* \leftarrow \mathbf{T}^{\text{init}}$
**while** not converged **do**
  $\mathbf{T}^{\text{old}} \leftarrow \mathbf{T}^*$
  $\mathcal{I} \leftarrow \texttt{nnsearch}(\mathcal{X}_s, \mathcal{X}_t, \mathbf{T}^{\text{old}})$ // Find Association
  $\mathbf{T}^* \leftarrow \arg\max_{\mathbf{T}\in\mathrm{SE}(3)} f_{\texttt{GICP}}(\mathbf{T}; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I})$ // Optimize over $\mathrm{SE}(3)$
  **if** $\mathrm{d}_{\mathrm{SE}(3)}(\mathbf{T}^{\text{old}}, \mathbf{T}^*) < \epsilon$ **then** // Check convergence using distance threshold $\epsilon$
    converged $\leftarrow$ true
  **end if**
**end while**
**return** $\mathbf{T}^*$

We will also use this metric in the evaluations. Figure 2.2 shows the convergence of GICP-$\mathrm{SE}(3)$ compared to that of GICP. The parametrization over $\mathrm{SE}(3)$ leads to convergence in fewer iterations versus Euler angles.

## 2.5 Semantic Iterative Closest Point

Problem 2 frames the semantic point cloud registration problem as an optimization of the joint log-likelihood. However, both semantic class, $\mathcal{S}$, and association, $\mathcal{I}$, are in fact latent variables. The domain of the semantic random variables are small in this work and we can directly marginalize them; unfortunately, the same does not hold for associations. Following an EM approach, the joint likelihood is $p(\mathcal{R}, \mathcal{S}, \mathcal{I}|\mathcal{X}_t, \mathcal{X}_s)$. We now assume, given the semantic class and association, the points $\mathbf{x}_k^t$ and $\mathbf{x}_k^s$ have independent noise, i.e., they are independent measurements. Together with applying Bayes' rule, it is easy to show that

$$p(\mathcal{R}, \mathcal{S}, \mathcal{I}|\mathcal{X}_t, \mathcal{X}_s) = p(\mathcal{R}|\mathcal{S}, \mathcal{I}, \mathcal{X}_t, \mathcal{X}_s) p(\mathcal{S}|\mathcal{I}, \mathcal{X}_t) p(\mathcal{S}|\mathcal{I}, \mathcal{X}_s) \frac{p(\mathcal{I}|\mathcal{X}_t, \mathcal{X}_s)}{p(\mathcal{S}|\mathcal{I})} \tag{2.4}$$

where $p(\mathcal{S}|\mathcal{I})$ can be seen as an uninformative prior term.

Similar to McCormac et al. (2017a), given the point cloud and association, we use a CNN to model the per point semantic observation term, $p(\mathcal{S}|\mathcal{I}, \mathcal{X})$. We model $p(\mathcal{R}|\mathcal{S}, \mathcal{I}, \mathcal{X}_t, \mathcal{X}_s)$ using the same Gaussian distribution as in GICP Segal, Haehnel, and Thrun (2009). Since the residual is a function of the $\mathbf{x}_k^t$, $\mathbf{x}_k^s$, and the association $i_k$, it is independent of the semantic class given those variables, and we can simplify $p(\mathcal{R}|\mathcal{S}, \mathcal{I}, \mathcal{X}_t, \mathcal{X}_s)$ to $p(\mathcal{R}|\mathcal{I}, \mathcal{X}_t, \mathcal{X}_s)$. Consequently, (2.4) simplifies to

$$p(\mathcal{R}, \mathcal{S}, \mathcal{I}|\mathcal{X}_t, \mathcal{X}_s) \propto \underbrace{p(\mathcal{R}|\mathcal{I}, \mathcal{X}_t, \mathcal{X}_s)}_{\text{residual}} \underbrace{p(\mathcal{S}|\mathcal{I}, \mathcal{X}_t)}_{\text{target semantic}} \underbrace{p(\mathcal{S}|\mathcal{I}, \mathcal{X}_s)}_{\text{source semantic}} \underbrace{p(\mathcal{I}|\mathcal{X}_t, \mathcal{X}_s)}_{\text{geometric association}} \tag{2.5}$$

Our approach in **Step 2** differs from Section 2.4 in how we handle the latent variable that represents the associations. The standard nearest neighbor approach can be seen as a heuristic of picking the geometrically closest point as a hard association. In contrast, we defined the geometric association, $p(\mathcal{I}|\mathcal{X}_t, \mathcal{X}_s) = \prod_{k=1}^{n} p(i_k|\mathbf{x}_k^t, \mathbf{x}_k^s)$, as

$$p(i_k|\mathbf{x}_k^t, \mathbf{x}_k^s) \triangleq \begin{cases} \frac{1}{N} & \text{if } \mathbf{x}_k^t \text{ is } N \text{ nearest neighbors of } \mathbf{x}_k^s \\ 0 & \text{otherwise} \end{cases} \tag{2.6}$$

The EM approach to infer the latent variables and the optimal transformation, $\mathbf{T}$, can be expressed as follows:

– Expectation: We wish to compute the expected value of the log-likelihood function with respect to the probability of the association given the current transformation and point clouds, or the $Q(\cdot, \cdot)$ function.

$$\begin{aligned} Q(\mathbf{T}, \mathbf{T}^{\text{old}}) &= \mathbb{E}_{p(\mathcal{I}|\mathcal{R},\mathcal{S},\mathcal{X}_t,\mathcal{X}_s;\mathbf{T}^{\text{old}})}[\log p(\mathcal{R}, \mathcal{S}, \mathcal{I}|\mathcal{X}_t, \mathcal{X}_s; \mathbf{T})] \\ &= \sum_{\mathcal{I}\in\mathbb{I}} p(\mathcal{I}|\mathcal{R}, \mathcal{S}, \mathcal{X}_t, \mathcal{X}_s; \mathbf{T}^{\text{old}}) \log p(\mathcal{R}|\mathcal{I}, \mathcal{X}_t, \mathcal{X}_s; \mathbf{T}) + \text{const.} \end{aligned} \tag{2.7}$$

– Maximization: We wish to maximize $Q(\cdot, \cdot)$ over the transformation variable $\mathbf{T}$

$$\mathbf{T}^* = \underset{\mathbf{T}\in\text{SE(3)}}{\arg\max}\, Q(\mathbf{T}, \mathbf{T}^{\text{old}}) \tag{2.8}$$

We can see that (2.7) is log-likelihood of (2.4). The probability of the latent variable given data and the current transformation estimate, $p(\mathcal{I}|\mathcal{R}, \mathcal{S}, \mathcal{X}_t, \mathcal{X}_s; \mathbf{T}^{\text{old}})$, can be expanded as

$$\begin{aligned} p(\mathcal{I}|\mathcal{R}, \mathcal{S}, \mathcal{X}_t, \mathcal{X}_s; \mathbf{T}^{\text{old}}) &= \frac{p(\mathcal{R}, \mathcal{S}, |\mathcal{I}, \mathcal{X}_t, \mathcal{X}_s, ; \mathbf{T}^{\text{old}}) p(\mathcal{I}|\mathcal{X}_t, \mathcal{X}_s)}{\sum_{\tilde{\mathcal{I}}\in\mathbb{I}} p(\mathcal{R}, \mathcal{S}|\tilde{\mathcal{I}}, \mathcal{X}_t, \mathcal{X}_s, ; \mathbf{T}^{\text{old}}) p(\tilde{\mathcal{I}}|\mathcal{X}_t, \mathcal{X}_s)} \\ &\triangleq \eta\, p(\mathcal{R}, \mathcal{S}|\mathcal{I}, \mathcal{X}_t, \mathcal{X}_s, ; \mathbf{T}^{\text{old}}) p(\mathcal{I}|\mathcal{X}_t, \mathcal{X}_s) \end{aligned} \tag{2.9}$$

where $\eta$ is constant with respect to $\mathcal{I}$. Using (2.5), we calculate a weight based on the conditional probability of every possible association, denoted by $i_k \in \mathbb{I}$, excluding the normalization constant $\eta$ and uninformative priors, as follows.

$$w_k \triangleq \sum_{s_k\in\mathcal{C}} p(\mathbf{r}_k|\mathbf{x}_k^t, \mathbf{x}_k^s, i_k; \mathbf{T}^{\text{old}}) p(s_k|\mathcal{X}_t, i_k) p(s_k|\mathcal{X}_s, i_k) p(i_k|\mathbf{x}_k^t, \mathbf{x}_k^s) \tag{2.10}$$

We combine the weights from (2.10) into a weight array, $\mathbf{w} = \text{vec}(w_1, \ldots, w_{n\times N})$, that is

**Algorithm 2** Semantic ICP

**Require:** Initial transformation $\mathbf{T}^{\text{init}}$, target point cloud $\mathcal{X}_t$, source point cloud $\mathcal{X}_s$, semantic labels;
1: $\mathbf{T}^* \leftarrow \mathbf{T}^{\text{init}}$
2: **while** not converged **do**
3: $\quad\mathbf{T}^{\text{old}} \leftarrow \mathbf{T}^*$
4: $\quad\mathbf{w} \leftarrow$ Compute weights using (2.10) // Expectation
5: $\quad\mathbf{T}^* \leftarrow \arg\max_{\mathbf{T}\in\text{SE}(3)} f_{\texttt{SICP}}(\mathbf{T}, \mathbf{w}; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I})$ // Maximization: Optimize over $\text{SE}(3)$
6: $\quad$**if** $\text{d}_{\text{SE}(3)}(\mathbf{T}^{\text{old}}, \mathbf{T}^*) < \epsilon$ **then** // Check convergence using distance threshold $\epsilon$
7: $\quad\quad\text{converged} \leftarrow \text{true}$
8: $\quad$**end if**
9: **end while**
10: **return** $\mathbf{T}^*$

$n \times N$ counting non-zero weights. Subsequently, the maximization step becomes

$$\mathbf{T}^* = \underset{\mathbf{T}\in\text{SE}(3)}{\arg\max}\, f_{\texttt{SICP}}(\mathbf{T}, \mathbf{w}; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) \triangleq \underset{\mathbf{T}\in\text{SE}(3)}{\arg\max} \sum_{k=1}^{n\times N} \rho_\alpha(w_k ||\mathbf{x}_k^t - \mathbf{T}(\mathbf{x}_k^s)||^2_{\mathbf{C}_k}) \tag{2.11}$$

The algorithmic implementation of semantic ICP is shown in Algorithm 2. The steps are similar to the presented GICP-$\text{SE}(3)$ and the main difference is in line 4 where the weights are calculated, turning 5 into a weighted non-linear least squares problem over $\text{SE}(3)$.

## 2.6 Evaluation

We evaluate the performance of our proposed method by comparing the relative transformation error of the algorithms on two open source datasets, namely the KITTI Vision benchmark dataset Geiger, Lenz, and Urtasun (2012), and SceneNet RGBD Dataset McCormac et al. (2017b). We use two different CNNs for semantic inference on these datasets as one shows good performance on outdoor images while the other on the indoor scenes in our experiments. We show how using semantics in association and optimizing over $\text{SE}(3)$ benefit the point cloud registration task by comparing the following algorithms: Generalized ICP Segal, Haehnel, and Thrun (2009) available in the Point Cloud Library Rusu and Cousins (2011), GICP-$\text{SE}(3)$ described in Section 2.4 and Algorithm 1, Semantic ICP as described in Section 2.5 and Algorithm 2, and finally, the EM approach in Iterative Probabilistic Data Association (IPDA) Gabriel Agamennoni and Sorrenti (2016).

Table 2.1: Parameters used for each algorithm, similar values were chosen when possible, with the exception of IPDA which has a slightly different framework, for which we stayed close to the parameters in the authors implementation (Fontana, Hinzmann, and Agamennoni, 2016).

| **Parameters** | Semantic ICP | GICP-$\mathrm{SE}(3)$ | GICP | IPDA |
|---|---|---|---|---|
| Convergence Threshold $\epsilon$ | $1\mathrm{e}{-5}$ | $1\mathrm{e}{-5}$ | $1\mathrm{e}{-5}$ | $1\mathrm{e}{-3}$ |
| Outer Max Iterations | 50 | 50 | 50 | 50 |
| Inner Max Iterations | 200 | 200 | 200 | 100 |
| Solver Backend | Ceres | Ceres | PCL | Ceres |
| Solver Algorithm | LM | LM | BFGS | LM |
| Jacobian | Analytical | Analytical | Analytical | Auto Diff |
| Parameter Representation | $\mathrm{SE}(3)$ | $\mathrm{SE}(3)$ | Euler Angles $\times\ \mathbb{R}^3$ | $\mathrm{SE}(3)$ |
| Number of Threads | 8 | 8 | 1 | 8 |
| Distribution NN | 20 | 20 | 20 | NA |
| EM NN | 4 | NA | NA | 4 |
| NN Dist. Threshold | NA | NA | 1.5 m | 1.5 m |
| Cauchy Loss $\alpha$ | 2.0 | 2.0 | NA | NA |

### 2.6.1 Optimization

This section will present the specifics of our evaluation, including parameters used by each algorithm and the hardware they were run on. Table 3.1 lists the parameters of each algorithm used. Notable difference includes the use of Ceres Solver's automatic differentiation by IPDA Fontana, Hinzmann, and Agamennoni (2016). The experiments were run with version 1.13 of Ceres Solver, 3.3.4 of the Eigen matrix library, and version 1.8.1 of the Point Cloud Library. Timing results are presented on a computer with an Intel Core i7-3770 CPU, Nvidia Titan X (Pascal) GPU, and 32 GB of RAM.

### 2.6.2 KITTI Visual Odometry Dataset

We use the KITTI visual odometry dataset Geiger, Lenz, and Urtasun (2012) to evaluate the accuracy of the estimated transformations on the stereo data available as part of the dataset. The semantic inference is performed using Dilation CNN Yu and Koltun (2016). Disparity maps are created by the LIBELAS algorithm presented by Geiger, Roser, and Urtasun (2010) using the rectified stereo images. For segmentation results, we used Dilation CNN Yu and Koltun (2016) with a model trained on the KITTI dataset. Table 2.2 shows the statistics of the CNN on the dataset. We ran our evaluation on sequence 5 of the dataset. For the results presented, we calculated a trajectory with each algorithm by aligning every third point cloud. We then compared these relative transformations to the ground truth trajectories provided in the dataset.

Table 2.2: Dilation CNN performance measure on the KITTI Odomentry Dataset

| | Global Acc | Class Average Acc | mIoU | Inference Time (ms/image) |
|---|---|---|---|---|
| **Dilation CNN** | 0.9738 | 0.9242 | 0.8482 | 214 |

Table 2.3: KITTI results with the distance metrics and runtime. Best results for each column are in **bold**.

| **Algorithm** | **Tranfomation Error** $\mathrm{d}_{\mathrm{SE}(3)}(\mathbf{T}^*, \mathbf{T}_{\mathrm{GT}})$ | | **Rotational Error** $\mathrm{d}_{\mathrm{SO}(3)}(\mathbf{T}^*, \mathbf{T}_{\mathrm{GT}})$ (deg) | | **Translation Error** $\mathrm{d}_{\mathbb{R}^3}(\mathbf{T}^*, \mathbf{T}_{\mathrm{GT}})$ (m) | | **Runtime** (s) | |
|---|---|---|---|---|---|---|---|---|
| | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| Semantic ICP | **0.2619** | **0.2078** | **0.2041** | 0.1561 | **0.2618** | **0.2078** | 109.4 | 101.5 |
| GICP-$\mathrm{SE}(3)$ | 0.4923 | 0.2295 | 0.2467 | **0.1308** | 0.4922 | 0.2295 | 38.6 | 36.5 |
| GICP | 0.9095 | 0.4860 | 0.4200 | 0.3264 | 0.9094 | 0.4869 | **12.5** | **12.0** |
| IPDA | 1.1808 | 0.8732 | 1.2830 | 0.8341 | 1.1798 | 0.8731 | 2672.0 | 2555.0 |

To model $p(s_k|\mathcal{X}, i_k)$, as in (2.10), we fit a generalized Bernouli distribution to the same NN used to fit the Guassian residual distribution. That gives a distribution over CNN labels, to get the distribution over the true semantic class, we take the vector-matrix product of that distribution with the normalized confusion matrix collected on the training data. That gives us another vector that is the generalized Bernouli distribution of true semantic class in that area.

We use the distance metric $\mathrm{d}_{\mathrm{SE}(3)}(\cdot, \cdot)$ to compare estimated transformations to the ground truth trajectory provided in the dataset. We also provide $\mathrm{d}_{\mathrm{SO}(3)}(\cdot, \cdot)$ and $\mathrm{d}_{\mathbb{R}^3}(\cdot, \cdot)$ distances in Table 2.3. We define those distances as $\mathrm{d}_{\mathrm{SO}(3)}(\mathbf{T}_1, \mathbf{T}_2) = \|\log(\mathbf{R}_1\mathbf{R}_2^\top)^\vee\|$ and $\mathrm{d}_{\mathbb{R}^3}(\mathbf{T}_1, \mathbf{T}_2) = \|\mathbf{t}_1 - \mathbf{R}_1\mathbf{R}_2^\top\mathbf{t}_2\|$ which are consistent with the $\mathrm{d}_{\mathrm{SE}(3)}(\cdot, \cdot)$ definition.

We were only able to run IPDA on a subset of the dataset due to its high processing time. Table 2.3 summarizes the quantitative results on that subset. It shows that changing the parametrization of the optimization improved the results. In addition, adding semantics to aid the association problem further improved the results. The IPDA algorithm performed similarly to GICP, and it would potentially perform better when used to align two point clouds of varying density. The table also includes average runtimes, which are slower for the EM-based approaches. This is expected because soft associations add a factor more terms (equal to the number of neighbors of each point considered) to the cost function summation.

In Figure 2.3, to show the distribution of errors, we also plot the cumulative distribution function and box plots of the $\mathrm{d}_{\mathrm{SE}(3)}(\cdot, \cdot)$ for each algorithm. The plots show that both Semantic ICP and GICP-$\mathrm{SE}(3)$ outperform GICP regarding their best two quartiles. We can also observe that

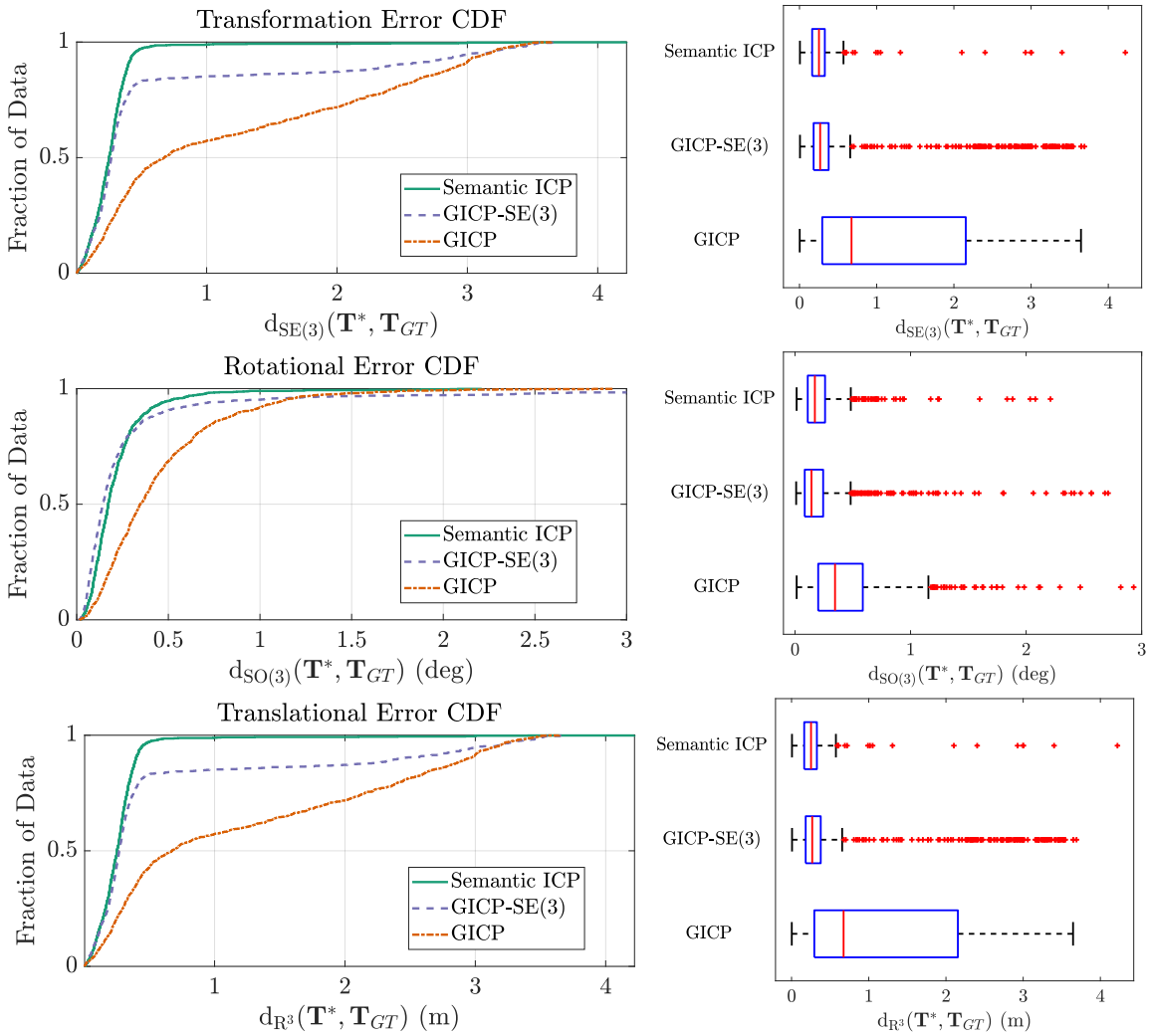



Figure 2.3: Error CDF and box plots of the proposed algorithms compared with GICP computed using KITTI sequence 05 dataset. The metrics used for comparison are $d_{SE(3)}(\cdot,\cdot)$, $d_{SO(3)}(\cdot,\cdot)$, $d_{\mathbb{R}^3}(\cdot,\cdot)$. The proposed algorithms, Semantic ICP and GICP-$SE(3)$, show better performance by exploiting the structure of $SE(3)$.

Semantic ICP starts to outperform GICP-$\mathrm{SE}(3)$ in its final quartile, which accounts for its better mean value.

Figure 2.4 shows the qualitative results of running Semantic ICP and GICP on a series of KITTI point clouds and then projecting them into a common reference frame. It shows that misalignment with GICP causes echos of objects, while Semantic ICP produces crisp point clouds.

Since all algorithms use gradient-based optimizers, the initialization affects the accuracy of results. To explore how our approach influences the basin of convergence, we plot the initial offset versus the final offset in Figure 2.5. We find that using $\mathrm{SE}(3)$ parametrization and incorporating semantic information improve convergence.

### 2.6.3 SceneNet RGBD Dataset

The SceneNet RGBD dataset is a synthetic rendered dataset that provides pixel level semantics and ground truth depth and camera trajectory McCormac et al. (2016, 2017b). The dataset was made by randomly generating indoor scenes and placing models of household objects in rooms. Random trajectories are then sampled and synthetic images are rendered. The dataset gives the ability to evaluate the compared registration algorithms to a known ground truth trajectory. For semantic inference, we trained DeepLab-ResNet Chen et al. (2016) on the SceneNet RGBD training data.

Evaluation was performed on the validation portion of the dataset. The dataset provides ground-truth depth, for evaluation we added independent Gaussian noise to each depth measurement $n_{\texttt{Depth}} \sim \mathcal{N}(0, (0.04m)^2)$. Four trajectories were used (29, 223, 530, and 784).

DeepLab system re-purposes image classification networks for semantic segmentation by applying atrous convolution with upsampling filters, and yields significant improvement over its baselines. We chose DeepLab-ResNet, which is built on a re-purposed ResNet-101 He et al. (2016), as the framework for semantic inference on SceneNet RGBD dataset. We initialized the model with weights pre-trained on MS-COCO dataset Lin et al. (2014), and fine-tuned it on the first training set (train_0) of the SceneNet RGBD dataset which includes 300000 images. The semantic annotations are obtained by mapping instance labels given in SceneNet RGBD to NYUv2 13 class semantic labels Couprie et al. (2013).

The network was trained using the standard stochastic gradient descent algorithm and the "poly" learning rate policy with the base learning rate set to 0.00025 and power to 0.9. Momentum and weight decay are set to 0.9 and 0.0005 respectively. We used a mini-batch size of 10, and trained the network for a total of 150K iterations for 3 days on an Nvidia TITAN X (Pascal). The performance of the network is shown in Table 2.4. For this network we modeled $p(s_k|\mathcal{X}, i_k)$ similarly to how we did for Subsection 2.6.2.

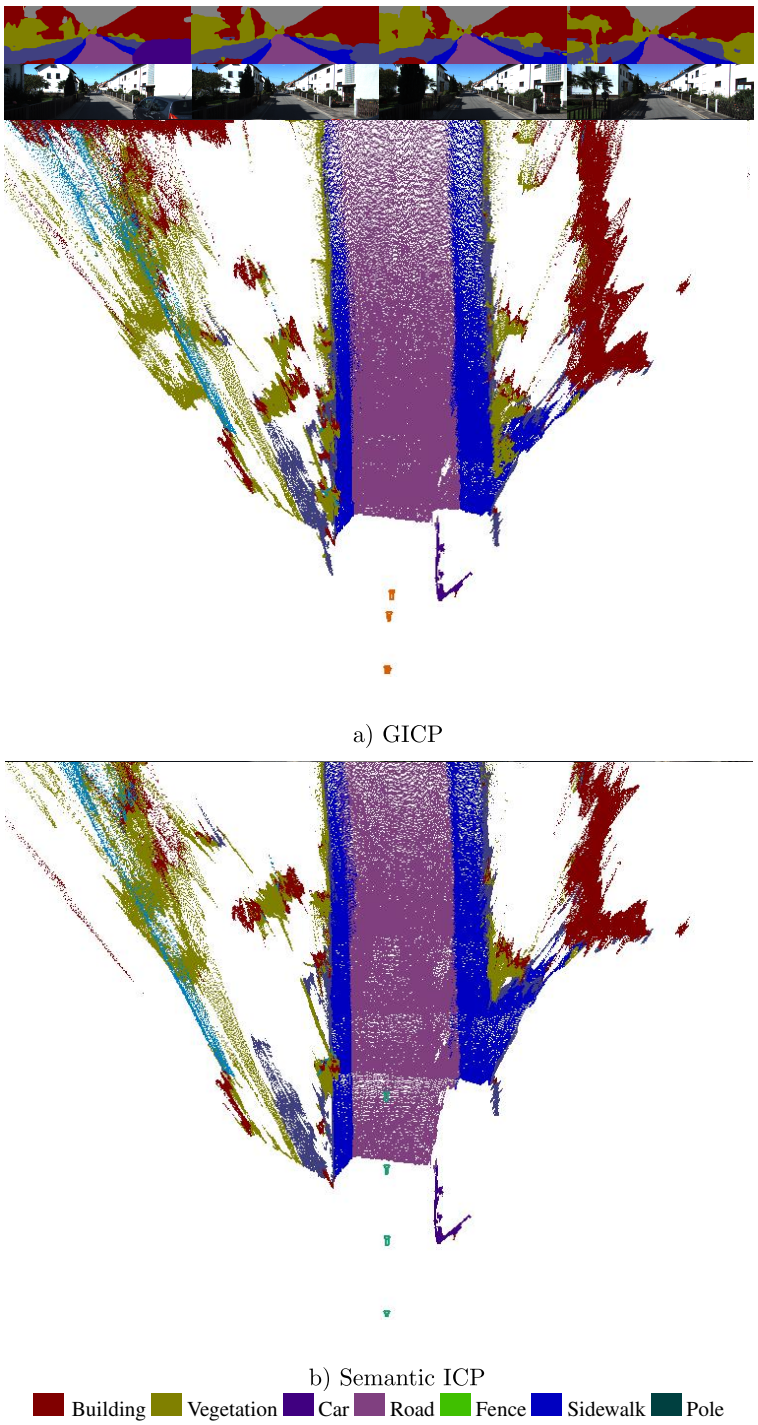



Figure 2.4: Sequential point clouds aligned using Semantic ICP on the right and GICP on the left. The top row shows the source image from the KITTI visual odometry dataset. The second row shows the inferred semantic labels produced using the Dilation CNN. The image on the left is the point clouds transformed by the estimated Semantic ICP transformations, with the camera positions marked in Cyan. The right are by the estimated GICP transformations with the camera positions marked in orange. The repeated object on the right side of the roadway are artifacts of poor alignment by GICP.

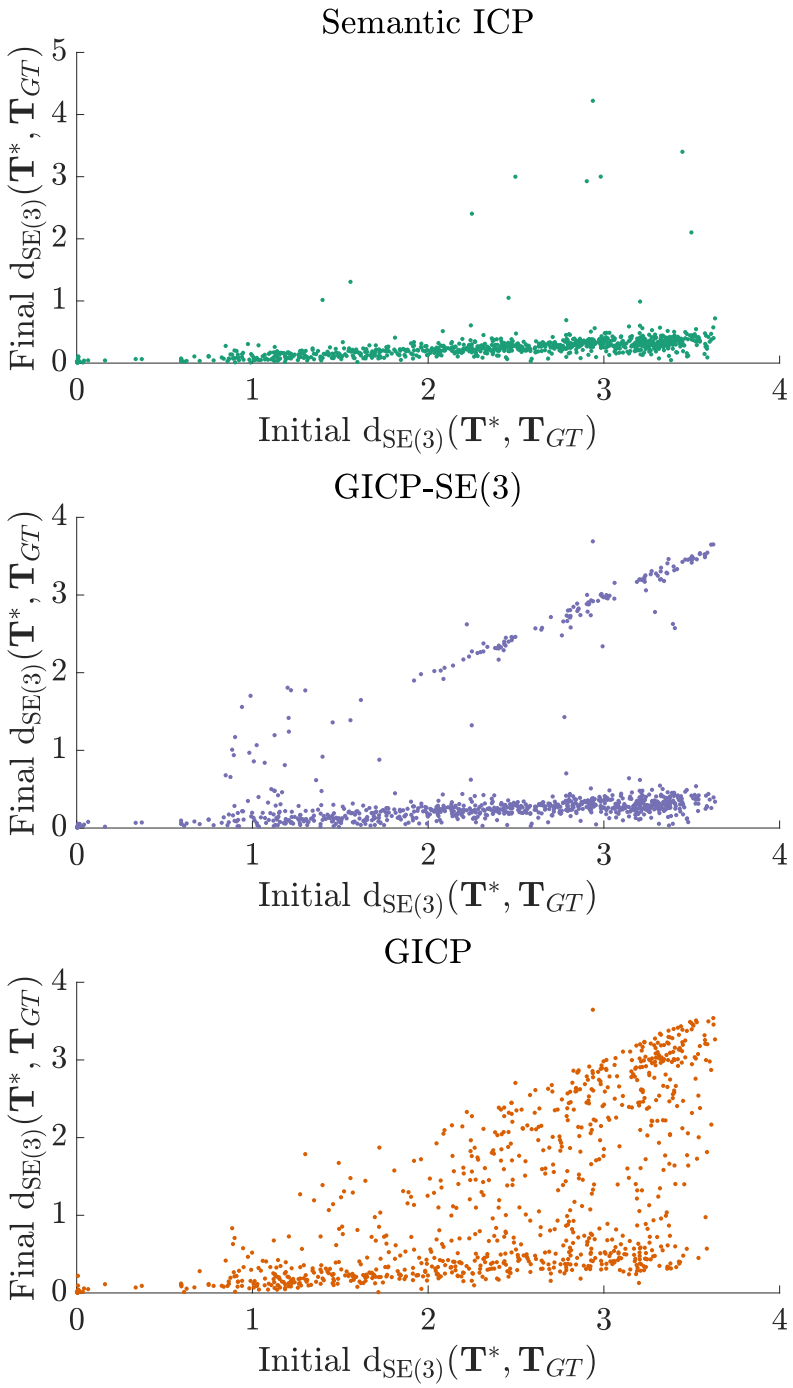



Figure 2.5: Scatter plots of the initial alignment vs. final alignment using $\mathrm{d}_{\mathrm{SE}(3)}(\cdot, \cdot)$ for each algorithm on the KITTI visual odometry dataset. We can see that GICP is less likely to converge as the initial offset gets larger, while Semantic ICP and GICP-SE(3) are more of a bimodal distribution, either staying near the initial transformation, or converging.

Table 2.4: DeepLab-ResNet performance measure on the SceneNet RGBD Dataset train_0

| | Global Acc | Class Average Acc | mIoU | Inference Time (ms/image) |
|---|---|---|---|---|
| **DeepLab-ResNet** | 0.8810 | 0.8453 | 0.7444 | 162 |

Table 2.5: SceneNet RGBD results with the distance metrics and runtime. Best result for each column is in **bold**.

| **Algorithm** | **Tranfomation Error** $\mathrm{d}_{\mathrm{SE(3)}}(\mathbf{T}^*, \mathbf{T}_{\mathrm{GT}})$ | | **Rotational Error** $\mathrm{d}_{\mathrm{SO(3)}}(\mathbf{T}^*, \mathbf{T}_{\mathrm{GT}})$ (deg) | | **Translation Error** $\mathrm{d}_{\mathbb{R}^3}(\mathbf{T}^*, \mathbf{T}_{\mathrm{GT}})$ (m) | | **Runtime** (s) | |
|---|---|---|---|---|---|---|---|---|
| | Mean | Median | Mean | Median | Mean | Median | Mean | Median |
| Semantic ICP | **0.4430** | **0.0377** | **9.98** | **0.5339** | **0.3778** | **0.0349** | 54.0 | 32.0 |
| GICP-$\mathrm{SE}(3)$ | 0.4602 | 0.0443 | 10.70 | 0.6874 | 0.3878 | 0.0425 | 15.2 | 9.0 |
| GICP | 0.4582 | 0.0629 | 10.29 | 1.04 | 0.3915 | 0.0598 | **2.8** | **2.0** |

Each algorithm was used to align consecutive point clouds in the dataset and $\mathrm{d}_{\mathrm{SE(3)}}(\cdot)$, $\mathrm{d}_{\mathrm{SO(3)}}(\cdot)$, and $\mathrm{d}_{\mathbb{R}^3}(\cdot)$ were collected with respect to the provided ground truth. The results are summarized in Table 2.5. The mean values are tighter than those of the KITTI visual odometry dataset, and larger than the medians, indicating a significant tail of errors. This is most likely caused by strong geometric features, such as perpendicular wall and ceiling, either being correctly associated or, in some outlier cases, completely miss-associated. Nevertheless, the Semantic ICP algorithm shows a quantitative improvement over GICP-$\mathrm{SE}(3)$.

Figure 2.6 shows error distance of the various methods in CDF and box plots. It shows a tighter grouping than was presented for the KITTI visual odometry dataset, but with Semantic ICP showing improvement over GICP and GICP-$\mathrm{SE}(3)$. Like the mean error metric, these plots are affected by the long tail of error values present in this data.

## 2.7 Conclusion

In this chapter, we proposed a novel algorithm for the point cloud registration problem that is based on the joint semantic and geometric inference. Our proposed Semantic ICP algorithm treats point associations as latent random variables leading to an EM-style solution. We showed semantic labels together with EM data associations improves the algorithm's performance in comparison with standard GICP and our GICP-$\mathrm{SE}(3)$. This evaluation was performed on two publicly available datasets. The extension of this work to a framework for semantic SLAM or an odometry system

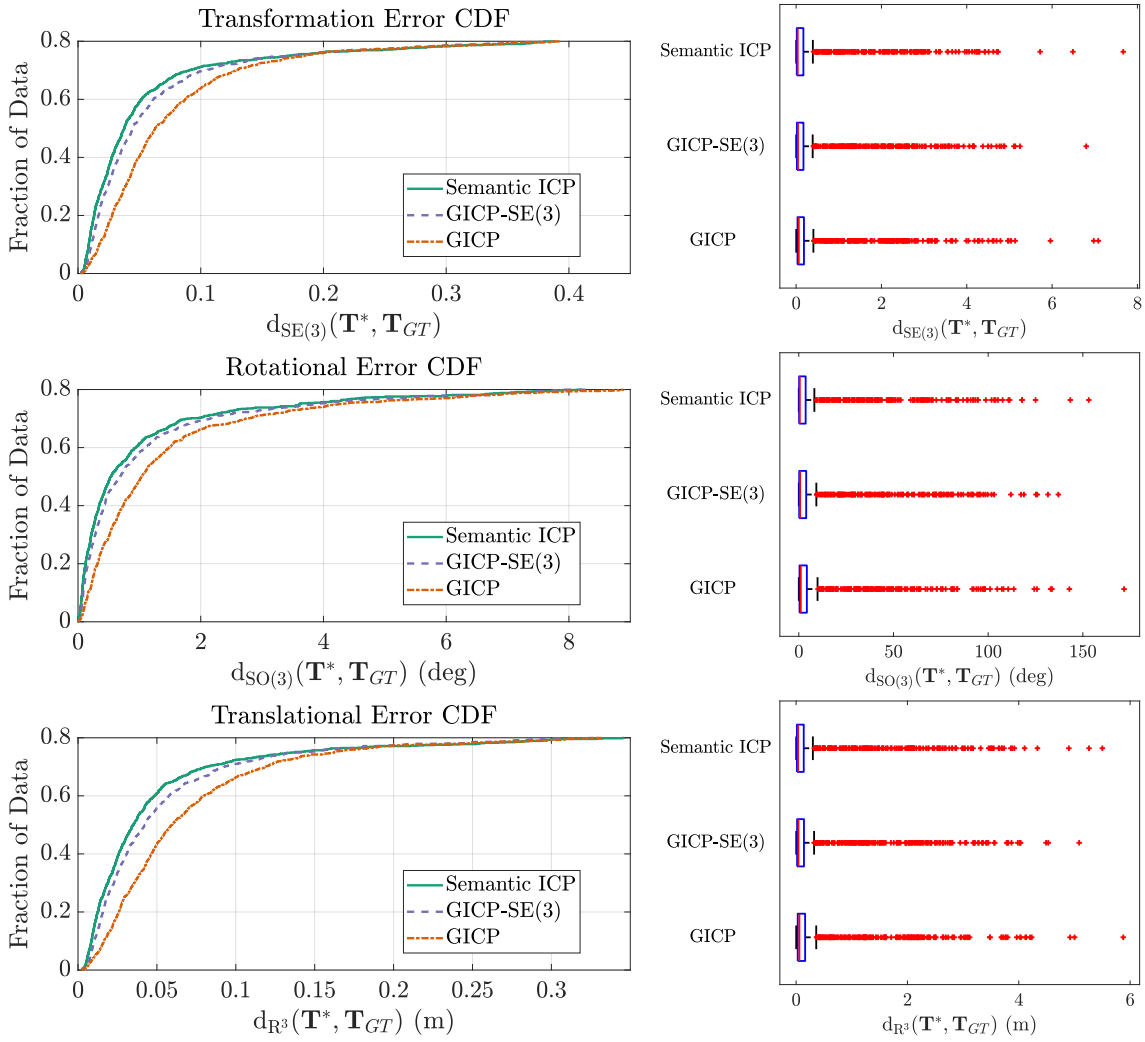



Figure 2.6: Box plots and cumulative distribution functions for the SceneNet RGBD dataset. There is a long tail on the errors for all methods for this dataset. The limited field of view and close-in objects led to high angle error. The CDFs are cut at 80% of the data to make the difference more clear but all data is visible in the box plots.

for real-time applications is an interesting future direction (Valencia et al., 2009; Wolcott and Eustice, 2017a). Extensions to optimizing over multiple rigid body transformations to compensate for dynamic objects in the scene is also an interesting direction.

# CHAPTER 3

# Boosting Shape Registration Algorithms via Reproducing Kernel Hilbert Space Regularizers

In this chapter we present a method for incorporating extra channel information into the registration process. In particular we are interested in transformation invariant channels such as color and reflectivity of light detection and ranging (LIDAR) measurements. Such features provide additional information that makes alignment easier. In our proposed approach we use a sparse Bayesian technique to learn extra channel functions for the target and source point cloud. We then sample the functions and use the distance between the functions as a regularize to the sensor registration problem. This approach minimizes the affects off data associations (and therefor mis-associations) on the result of the registration problem.

## 3.1 Introduction

The shape registration problem is formulated as finding a rigid body transformation that aligns a set of source points to a set of target points. Registration algorithms can be divided into coarse alignment methods and fine alignment methods. Coarse alignment methods usually do not assume large overlap nor need an initial transformation, but only achieve a crude registration. In Makadia, Patterson, and Daniilidis (2006), a Fourier-based method is proposed to estimate the rotation of limited overlap point clouds. Recently, a deep neural network is used to encode local 3D geometric structures for coarse registration Elbaz, Avraham, and Fischer (2017). This chapter focuses on the fine registration problem.

Most modern fine alignment methods are derived from the Iterative Closest Point (ICP) algorithm, developed by Besl and McKay (1992). ICP iterates between finding the closest pair of points between the two sets of points, and minimizing the sum of geometric residuals between them. The ICP algorithm is extended to minimize point to line by Censi (2008), and point to plane residuals by Chen and Medioni (1991b).

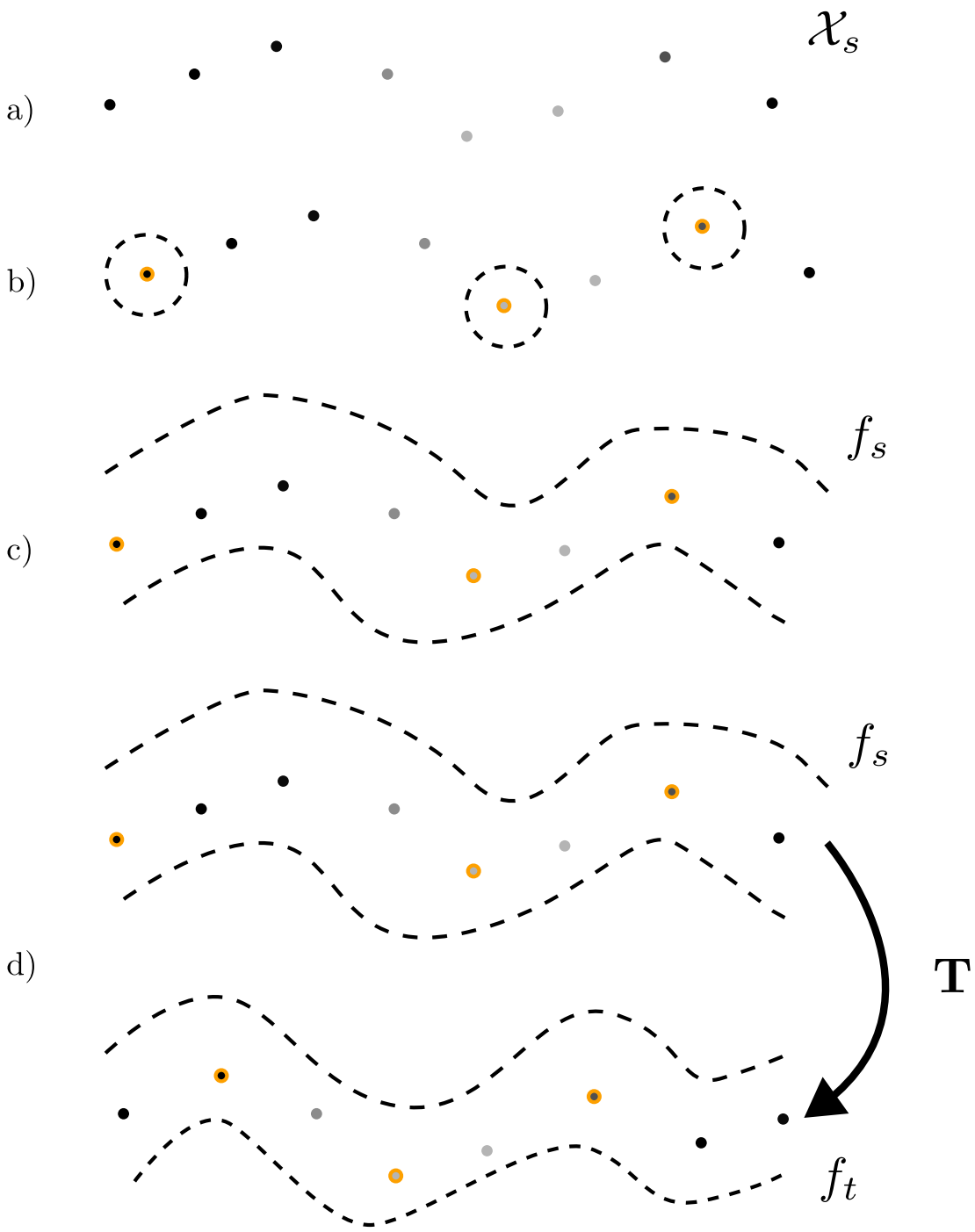



Figure 3.1: An illustration of the proposed regularization method. a) Shows the source point cloud $\mathcal{X}_s$ with intensity values. b) illustrates the first stage of our approach in which we train relevance vectors to approximate the intensity function $f_s$ in c). Finally in d) we minimize the regularized cost function to find the optimal transformation $\mathbf{T}$. We argue that the proposed RKHS regularizer is a natural regularizer for the registration problem at hand as it is agnostic to the choice of the registration cost function and is applicable to both LIDAR and RGB-D camera measurements.

These geometric interpretations of the registration problem have been extended to probabilistic frameworks. In the Generalized Iterative Closest Point (GICP) algorithm by Segal, Haehnel, and Thrun (2009), a Gaussian distribution is fit to the neighboring points of every point in each point cloud. Meanwhile, pairs of points whose residuals go beyond a hard threshold will be discarded. The normal distributions transform (NDT) algorithm by Stoyanov et al. (2012) divides $\mathbb{R}^3$ into voxels and fits a Gaussian distribution to all the points that fall into each voxel. Both algorithms then minimize the (Gaussian) distribution-to-distribution distance between the target and the source point clouds.

Purely geometric registration methods, such as the ones mentioned previously, ignore additional information obtained by the sensors, such as RGB (color information) or intensity values. We propose to use these additional channels of information to regularize the inherently geometric registration problem. This idea is not new and others have used RGB and intensity channels in the registration problem. Johnson and Kang (1999) added color to the nearest neighbor search to find the point minimizing the 6D distance, and then the 3D geometric distance between those points is minimized. Color Supported GICP by Korn, Holzkothen, and Pauli (2014) also adds color to the nearest neighbor search but defines the distance using the CIELAB color space instead of RGB. In Multi-Chanel GICP by Servos and Waslander (2017) a Gaussian distribution is fit over the Euclidean position parameters and the color channels of each point and the same distribution-to-distribution cost function by Segal, Haehnel, and Thrun (2009) is minimized. It also uses the extra channels in association by constructing a higher dimensional $k$-dimensional (KD) tree. In Color-NDT by Huhle et al. (2008) a Gaussian mixture model is constructed from the color channels of the points that fall into a voxel.

In the application of autonomous vehicles, intensity values from LIDAR have been used for online localization of a platform vehicle through registration (Levinson, Montemerlo, and Thrun (2007); Levinson and Thrun (2010)). First, an orthographic map of LIDAR intensity is generated as *a priori* using a simultaneous localization and mapping (SLAM) pipeline. Then, during online operation, intensity observations from LIDAR are used to register current observations into the map prior. This idea can be generalized for applications with other sensory modalities, such as cameras (Wolcott and Eustice (2014)).

Recent work towards direct visual odometry has led to a different approach to the registration problem using color or intensity based on the photo-consistency assumption. Instead of minimizing geometric residuals, these methods minimize photometric errors (Kerl, Sturm, and Cremers (2013); Engel, Stueckler, and Cremers (2015)). This is done by reprojecting the source points into the image frame in which the target points were captured and then minimizing the difference between RGB or intensity values. However, outliers caused by brightness changes do exist across different frames. Kerl, Sturm, and Cremers (2013) introduce a customized sensor model, t-distribution, to

model the error distribution, compensating the frequency of very large or very small photometric residuals.

Optical flow and scene flow approaches use similar methods to estimate relative motion. In optical flow, pixel-wise 2D relative motion between a pair of images can be estimated by leveraging an appearance-based constancy metric, such as brightness constancy, in an energy-minimization framework (Horn and Schunck (1981); Lucas and Kanade (1981); Liu et al. (2008); Barnes et al. (2009); Hu, Song, and Li (2016)). Scene flow tackles a similar problem in 3D with the use of a stereo camera system or active depth sensing (Vedula et al. (1999); Isard and MacCormick (2006); Ferstl et al. (2014) Hornacek, Fitzgibbon, and Rother (2014); Jaimez et al. (2015); Yan and Xiang (2016)).

Several methods have been developed that use the semantic label output of classifiers using intensity and RGBD point clouds as input. In the work by Zaganidis et al. (2018) semantic-assisted NDT, which restricts associates to points in the same class, and semantic-assisted GICP, which also restricts associations to the same class and computes the local covariance used in GICP using points of the same class, were introduced. A soft approach based on GICP and using expectation maximization with semantic probability distributions for associations was introduced by Parkison et al. (2018).

### 3.1.1 Contributions

We develop a novel class of regularizers modeled in the Reproducing Kernel Hilbert Space (RKHS) that ensures correspondences are also consistent in an abstract vector space of functions such as intensity surface, illustrated in Figure 3.1. The contributions of this work are as follows:

1. assuming the local consistency of point cloud intensity, we develop a class of regularizers to the Generalized-ICP registration algorithm over $\mathrm{SE}(3)$. To account for possible mismatches during data association, instead of using the difference of intensity directly, we learn the point cloud intensity function from noisy intensity measurements;

2. the open source implementation of the developed method including the registration and regression algorithms[1]

3. we evaluate the proposed method using publicly available experimental data and show the performance relative to related baselines.

[1] https://bitbucket.org/saparkison/rkhs_gicp

### 3.1.2 Outline

Appendix A provides the required preliminaries and notation. The problem formulation is given in Section 3.2. Section 3.3 discusses our main result on a class of regularized shape registration algorithms and an instance of RKHS regularization via sparse Bayesian inference algorithms. Section 3.4 presents the empirical results on LIDAR and RGB-D sensor data. Finally, Section 3.5 concludes the chapter and discusses future research directions.

### 3.1.3 Representation and Reproducing Kernel Hilbert Space

A Hilbert space is a complete inner product space. Let $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$ be a real Hilbert space of functions with the inner product between any two square-integrable functions $f, g \in \mathcal{H}$ (or $f, g \in L^2(\mathbb{R}, \mu)$) defined as:

$$\langle f, g \rangle_{\mathcal{H}} \triangleq \int f(\mathbf{x}) g(\mathbf{x}) \mathrm{d}\mu(\mathbf{x}), \tag{3.1}$$

where $\mu$ is the Lebesgue measure on $\mathbb{R}$. The induced norm by the inner product is $\|f\|_{\mathcal{H}} = \sqrt{\langle f, f \rangle_{\mathcal{H}}}$.

**Definition 3** (Reproducing Kernel Hilbert Space Berlinet and Thomas-Agnan (2004))**.** *Let $\mathcal{H}$ be a real-valued Hilbert space on a non-empty set $\mathcal{X}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of the Hilbert space $\mathcal{H}$ iff:*

1. $\forall \mathbf{x} \in \mathcal{X}, \quad k(\cdot, \mathbf{x}) \in \mathcal{H}$,
2. $\forall \mathbf{x} \in \mathcal{X}, \quad \forall f \in \mathcal{H} \quad \langle f, k(\cdot, \mathbf{x}) \rangle_{\mathcal{H}} = f(\mathbf{x})$.

*The Hilbert space $\mathcal{H}$ ($\mathcal{H}_k$) which possesses a reproducing kernel $k$ is called a Reproducing Kernel Hilbert Space or a proper Hilbert space.*

The second property is called the reproducing property; that is using the inner product of $f$ with $k(\cdot, \mathbf{x})$, the value of function $f$ is reproduced at point $\mathbf{x}$. There is a one-to-one relation between a reproducing kernel and its associated RKHS, and such a reproducing kernel is unique Berlinet and Thomas-Agnan (2004). Therefore, our problem reduces to finding an appropriate kernel.

Finally, the nonparametric representer theorem Schölkopf, Herbrich, and Smola (2001) ensures that the solution of minimizing the regularized risk functional admits a representation of the form

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, \mathbf{x}_i). \tag{3.2}$$

## 3.2 Problem Statement and Formulation

We wish to find the 3D rigid body transformation that aligns two point clouds. We use $\mathcal{X} \subset \mathbb{R}^3$ to denote a set of spatial coordinates returned by a range sensor. The following definitions are useful throughout the chapter.

**Definition 4** (Target point cloud)**.** *The point cloud $\mathcal{X}_t$ which is considered to be in a fixed reference frame is called the target point cloud.*

**Definition 5** (Source point cloud)**.** *The point cloud $\mathcal{X}_s$ which $\mathbf{T} \in \mathrm{SE}(3)$ acts on is called the source point cloud.*

**Definition 6** (Target function)**.** *Let $\mathcal{X}_t \subset \mathbb{R}^3$ be a point cloud which is considered in the fixed reference frame. The function $f_t : \mathcal{X}_t \to \mathbb{R}$ is called the target function.*

**Definition 7** (Source function)**.** *Let $\mathcal{X}_s \subset \mathbb{R}^3$ be a point cloud which $\mathbf{T} \in \mathrm{SE}(3)$ acts on it. The function $f_s : \mathcal{X}_s \to \mathbb{R}$ is called the source function.*

The target and source functions, in general, can represent any maps. For example, we can learn a function that maps a 3D point to intensity or curvature. In this work, we only consider intensity as the output of the regression since both stereo cameras and LIDARs directly provide such measurements associated with each point in the point cloud. In addition, the intensity measurements are also well-defined on sparse areas of point clouds, unlike curvature.

The action of $\mathbf{T}$ on any point $\mathbf{x}_i \in \mathcal{X}$ is $\mathbf{T} \cdot \mathbf{x}_i = \mathbf{R}\mathbf{x}_i + \mathbf{p}$, where $\mathbf{R} \in \mathrm{SO}(3)$ and $\mathbf{p} \in \mathbb{R}^3$. The likelihood function for aligning two point clouds sampled from the same environment depends on data association between them. We define the association variable $\mathcal{I} \triangleq \{i_k, j_k\}_{k=1}^n \in \mathbb{I}$ where $i_k, j_k$ indicate $\mathbf{x}_k^t \triangleq \mathbf{x}_{i_k}^t \in \mathcal{X}_t$ is a measurement of the same point as $\mathbf{x}_k^s \triangleq \mathbf{x}_{j_k}^s \in \mathcal{X}_s$, and $\mathbb{I}$ is the set of all possible associations (permutations). The association set $\mathcal{I}$ gives the indices of points in the target and source point clouds which are independent measurements of the same point. We also introduce a new random variable, $\mathcal{R} \triangleq \{\mathbf{r}_k\}_{k=1}^n$, to represent the residual where $\mathbf{r}_k \triangleq \mathbf{x}_k^t - \mathbf{T} \cdot \mathbf{x}_k^s$. To emphasize that the likelihood term includes the action of $\mathbf{T} \in \mathrm{SE}(3)$ on $\mathcal{X}_s$, we shall write the negative log-likelihood function as $\mathsf{cost}(\mathbf{T}) = \mathsf{cost}(\mathbf{T}; \mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}) \triangleq -\log p(\mathcal{R}|\mathcal{X}_t, \mathcal{X}_s, \mathcal{I}; \mathbf{T})$.

The ICP approach follows an iterative two-step procedure for solving the point cloud registration problem: 1) determine the association $\mathcal{I}$ using a Nearest Neighbor (NN) search; 2) minimize the cost defined using the residual, $\mathcal{R}$, over the parameter $\mathbf{T}$. In this work, we use a variant of GICP Segal, Haehnel, and Thrun (2009) that we call GICP-$\mathrm{SE}(3)$ Parkison et al. (2018); the main difference of GICP-$\mathrm{SE}(3)$ is solving the optimization on $\mathrm{SE}(3)$ rather than Euler angles parametrization which improves the convergence, and using a Cauchy loss function for robust estimation which removes the need for setting a commonly used distance threshold to accept or reject nearest neighbor data associations.

**Problem 3** ($\mathcal{H}_k$-regularized shape registration)**.** *Let $\mathcal{X}_t$ and $\mathcal{X}_s$ be two geometric point clouds and $f_t$ and $f_s$ be target and source functions learned using intensity measurements of their corresponding point clouds, respectively. Given correspondences between target and source point clouds, the optimal transformation that aligns source to target can be computed by solving the following regularized Maximum Likelihood Estimation (MLE) problem:*

$$\underset{\mathbf{T}\in \mathrm{SE}(3)}{\text{minimize}} \ \mathsf{cost}(\mathbf{T}) + \mathsf{reg}(\mathbf{T})$$

Without loss of generality, suppose we learn the target and source functions using the intensity measurements of their corresponding point clouds. Assuming the target and source functions are locally consistent and produce the same output for the corresponding inputs on the overlapping domain, we have $f_t(\mathbf{x}_k^t) = f_s(\mathbf{x}_k^s) = f_t(\mathbf{T} \cdot \mathbf{x}_k^s)$. To compute the distance between the target and source functions, we can use the induced norm in the corresponding RKHS as follows.

$$\|f_t - f_s\|_{\mathcal{H}_k}^2 = \left(f_t(\mathbf{T} \cdot \mathbf{x}_k^s) - f_s(\mathbf{x}_k^s)\right)^2. \tag{3.3}$$

Adding this equality constraint to the original problem and using the method of Lagrange multipliers, we arrive at the regularized shape registration problem, as shown in Problem 3. Further, we define the regularizer term as

$$\mathsf{reg}(\mathbf{T}) \triangleq \lambda \sum_{k=1}^{n} \left(f_t(\mathbf{T} \cdot \mathbf{x}_k^s) - f_s(\mathbf{x}_k^s)\right)^2. \tag{3.4}$$

## 3.3 A Class of $\mathcal{H}_k$-Regularized Shape Registration Algorithms

We model measurements in the target and source clouds as being drawn from Gaussian distributions, i.e., $\mathbf{x}_k^t \sim \mathcal{N}(\hat{\mathbf{x}}_k^t, \mathbf{\Sigma}_k^t)$, and $\mathbf{x}_k^s \sim \mathcal{N}(\mathbf{T} \cdot \hat{\mathbf{x}}_k^s, \mathbf{\Sigma}_k^s)$, respectively. Therefore, the residual log-likelihood, excluding the normalization constant, becomes:

$$\mathsf{cost}(\mathbf{T}) = \sum_{k=1}^{n} \|\mathbf{x}_k^t - \mathbf{T} \cdot \mathbf{x}_k^s\|_{\mathbf{C}_k}^2, \tag{3.5}$$

where $\mathbf{C}_k \triangleq \mathbf{\Sigma}_k^t + \mathbf{R}\mathbf{\Sigma}_k^s\mathbf{R}^\mathsf{T}$. The analytical gradient of this cost function in the ambient Euclidean space with respect to the translation and rotation, respectively, are

$$\frac{\partial \mathsf{cost}}{\partial \mathbf{p}} = \sum_{k=1}^{n} -2\mathbf{C}_k^{-1}\mathbf{r}_k, \ \frac{\partial \mathsf{cost}}{\partial \mathbf{R}} = \sum_{k=1}^{n} -2\mathbf{C}_k^{-1}\mathbf{r}_k(\mathbf{x}_k^{s\mathsf{T}} + \mathbf{r}_k^\mathsf{T}\mathbf{C}_k^{-1}\mathbf{R}\mathbf{\Sigma}_k^s).$$

The original GICP Segal, Haehnel, and Thrun (2009) removes residuals larger than a certain

value to ensure that any point in the source cloud which does not have a counterpart will not affect the solution. To avoid having a hard threshold, we replace this step with a robust estimator using the Cauchy loss function, $\rho_\alpha(x) = \alpha^2 \ln(1 + \frac{x}{\alpha^2})$, where $\alpha$ is a parameter that controls where the loss begins to scale sublinearly. Similar to the approach in GICP, the robust estimator diminishes the effect of outliers while avoiding the removal of potential inliers. Consequently, our cost function becomes:

$$\mathsf{cost}_\rho(\mathbf{T}) \triangleq \sum_{k=1}^{n} \rho_\alpha(||\mathbf{x}_k^t - \mathbf{T} \cdot \mathbf{x}_k^s||^2_{\mathbf{C}_k}), \tag{3.6}$$

and the effect of the loss function on the gradient is trivial to derive using the chain rule.

Following (3.2), the functions follows a representation such as $f_t(\cdot) = \sum_{j=1}^{m} \beta_j k_{\mathrm{SE}}(\cdot, \mathbf{z}_j)$. The Squared Exponential (SE) kernel has the form: $k_{\mathrm{SE}}(\mathbf{x}, \mathbf{z}) = \sigma_f^2 \exp(-\|\mathbf{x} - \mathbf{z}\|^2_{\mathbf{L}})$ where $\mathbf{L}$ is the diagonal matrix of characteristic length-scales and $\sigma_f^2$ is the signal variance. This is the most common kernel used in regression techniques using kernel methods Schölkopf and Smola (2002) such as Gaussian processes Rasmussen and Williams (2006) and Relevance Vector Machine (RVM) Tipping (2001), and we choose it as part of the model selection due to its smoothness and being infinitely differentiable. Consequently, the regularizer term becomes

$$\mathsf{reg}(\mathbf{T}) = \lambda \sum_{k=1}^{n} (\sum_{j=1}^{m} \beta_j k_{\mathrm{SE}}(\mathbf{T} \cdot \mathbf{x}_k^s, \mathbf{z}_j) - f_s(\mathbf{x}_k^s))^2. \tag{3.7}$$

The analytical gradients of this term in the ambient Euclidean space with respect to the translation and rotation, respectively, are

$$\frac{\partial \mathsf{reg}}{\partial \mathbf{p}} = \lambda \sum_{k=1}^{n} \sum_{j=1}^{m} a_k \beta_j k_{\mathrm{SE}}(\mathbf{z}_j, \mathbf{T} \cdot \mathbf{x}_k^s) \mathbf{L}^{-1} (\mathbf{z}_j - \mathbf{T} \cdot \mathbf{x}_k^s),$$

$$\frac{\partial \mathsf{reg}}{\partial \mathbf{R}} = \lambda \sum_{k=1}^{n} \sum_{j=1}^{m} a_k \beta_j k_{\mathrm{SE}}(\mathbf{z}_j, \mathbf{T} \cdot \mathbf{x}_k^s) \mathbf{L}^{-1} (\mathbf{z}_j - \mathbf{T} \cdot \mathbf{x}_k^s) \mathbf{x}_k^{s\mathsf{T}},$$

where $a_k \triangleq -2 \left[ \sum_{j=1}^{m} \beta_j k_{\mathrm{SE}}(\mathbf{T} \cdot \mathbf{x}_k^s, \mathbf{z}_j) - f_s(\mathbf{x}_k^s) \right]$.

### 3.3.1 $\mathcal{H}_k$-Regularization via Sparse Bayesian Inference

Given a training set $\mathcal{D} \triangleq \{\mathbf{x}_i, t_i\}_{i=1}^{n_t}$, $t_i$ is the noisy measurement (here intensity) of the real-valued latent $y_i$ for the input $\mathbf{x}_i \in \mathbb{R}^3$ (from point cloud), we model the functions with a linear model, $y(\mathbf{x}; \mathbf{w})$, as

$$y(\mathbf{x}; \mathbf{w}) \triangleq \sum_{j=1}^{n_b} w_j \phi_j(\mathbf{x}) = \mathbf{w}^\mathsf{T} \boldsymbol{\phi}(\mathbf{x}), \tag{3.8}$$

**Algorithm 3** $\mathcal{H}_k$-Regularized Shape Registration

**Require:** Initial transformation $\mathbf{T}^{\text{init}}$, target point cloud $\mathcal{X}_t$, source point cloud $\mathcal{X}_s$, optionally target function $f_t$;
$\mathbf{T}^{\text{OPT}} \leftarrow \mathbf{T}^{\text{init}}$ ▷ Initialize the transformation, e.g., $\mathbf{I}_4$
$f_s \leftarrow \texttt{rvm\_train}(\mathcal{X}_s)$ ▷ Target values are corresponding intensity measurements.
**if** $f_t$ not provided **then** ▷ In sequential data, the previous source function is the new target function.
  $f_t \leftarrow \texttt{rvm\_train}(\mathcal{X}_t)$
**end if**
converged $\leftarrow$ false
**while** not converged **do**
  $\hat{\mathbf{T}} \leftarrow \mathbf{T}^{\text{OPT}}$
  $\mathcal{I} \leftarrow \texttt{nnsearch}(\mathcal{X}_s, \mathcal{X}_t, \hat{\mathbf{T}})$ ▷ Find Association using NN search
  $\mathbf{T}^{\text{OPT}} \leftarrow \arg\min_{\mathbf{T} \in \text{SE}(3)} \mathsf{cost}_\rho(\mathbf{T}) + \lambda \mathsf{reg}(\mathbf{T})$ ▷ Optimize over SE(3)
  **if** $\mathrm{d}(\hat{\mathbf{T}}, \mathbf{T}^{\text{OPT}}) < \epsilon$ **then** ▷ Check convergence using distance threshold $\epsilon$
    converged $\leftarrow$ true
  **end if**
**end while**
**return** $\mathbf{T}^{\text{OPT}}$

where $\boldsymbol{\phi}_i \triangleq \boldsymbol{\phi}(\mathbf{x}_i) = \text{vec}(1, k_{\text{SE}}(\mathbf{x}_1, \mathbf{x}_i) \ldots, k_{\text{SE}}(\mathbf{x}_{n_b}, \mathbf{x}_i))$ are nonlinear basis functions. The weight vector, $\mathbf{w}$, is the model parameter whose distribution and dimension, $n_b$, to be learned Bishop (2006); Tipping (2004). We note that the choice of this model is justified by the representation in (3.2).

The objective is to infer $\mathbf{w}$ such that $y(\mathbf{x}; \mathbf{w})$ generalizes well to new inputs $\mathbf{x}_*$ (test data). In this work, we use RVM Tipping (2001) for the regression method. The sequential inference algorithm available for the RVM allows the method to be scalable while only a few (denoted by $n_b$ here) basis functions with non-zero weights survive (relevance vectors) in the final model, resulting in a sparse model.

The likelihood and the weight prior are modeled as Gaussian distributions. As a consequence of Gaussian likelihood and prior, the weight posterior, $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, can be computed in closed-form as follows.

$$\boldsymbol{\Sigma} = (\mathbf{A} + \sigma^{-2} \boldsymbol{\Phi}^{\mathsf{T}} \boldsymbol{\Phi})^{-1}, \tag{3.9}$$

$$\boldsymbol{\mu} = \sigma^{-2} \boldsymbol{\Sigma} \boldsymbol{\Phi}^{\mathsf{T}} \mathbf{t}, \tag{3.10}$$

where $\sigma^2$ is the likelihood variance, $\mathbf{A} \triangleq \text{diag}(\alpha_1, \ldots, \alpha_{n_b})$, the hyperparameters, $\alpha_1, \ldots, \alpha_{n_b}$, are the inverse variances of the weight priors, and $\boldsymbol{\Phi} \triangleq [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_{n_t}]^{\mathsf{T}}$.

For implementation, we followed the original software provided by Mike Tipping [2].

[2] http://www.miketipping.com/downloads.htm

Table 3.1: Parameters used for our algorithm on each dataset, similar values were chosen when possible or tuned on the same sequences. Parameters of the benchmark algorithms are reported in the software repository.

| $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ **Parameters** | KITTI | TUM RGB-D |
|---|---|---|
| Convergence Threshold $\epsilon$ | $1\mathrm{e}{-4}$ | $1\mathrm{e}{-4}$ |
| Outer Max Iterations | 50 | 50 |
| Inner Max Iterations | 100 | 100 |
| Solver Backend | Ceres Solver | Ceres Solver |
| Solver Algorithm | CG | CG |
| Jacobian | Analytical | Analytical |
| Parameter Representation | $\mathrm{SE}(3)$ | $\mathrm{SE}(3)$ |
| Distribution NN | 20 | 20 |
| Cauchy Loss $\alpha$ | 9.0 | 2.0 |
| Regularizer coefficient $\lambda$ | 20 | 5.0 |
| Kernel signal variance $\sigma_f^2$ | 12.5 | 2.5 |
| RVM Training Iterations | 200 | 200 |

### 3.3.2 Algorithmic Implementation

Algorithm 3 shows our implementation for solving the regularized form of the registration problem. In line 2 we learn the parameters of $f_s(\cdot)$ by maximizing the marginal likelihood, as presented in Subsection 3.3.1, following the sequential approach presented in Tipping, Faul et al. (2003). The regularization term is added to the cost function in line 10. Finding the association is done using a NN search in line 9. The optimization is solved over $\mathrm{SE}(3)$ using the Conjugate-Gradient solver in the open source optimization library Ceres Solver Agarwal, Mierle, and Others.

## 3.4 Experimental Results

We now evaluate the proposed algorithm using LIDAR and RGB-D sensors. We use GICP-$\mathrm{SE}(3)$ Parkison et al. (2018); Segal, Haehnel, and Thrun (2009) as the baseline for comparison, since it is the algorithm we applied our regularizer too. We also compare to NDTStoyanov et al. (2012), another method that does not use intensity or color, and Multichannel GICP (MC-GICP), a method that incorporates extra information into both the association and cost functionServos and Waslander (2017). For datasets where RGB data is available, we also compared to Color Supported GICP (GICP 6D) Korn, Holzkothen, and Pauli (2014). This method uses the distance in the CIELAB space to search for nearest neighbors. For NDT and GICP 6D we used the open source implementations available in the Point Cloud LibraryRusu and Cousins (2011), while for MC-

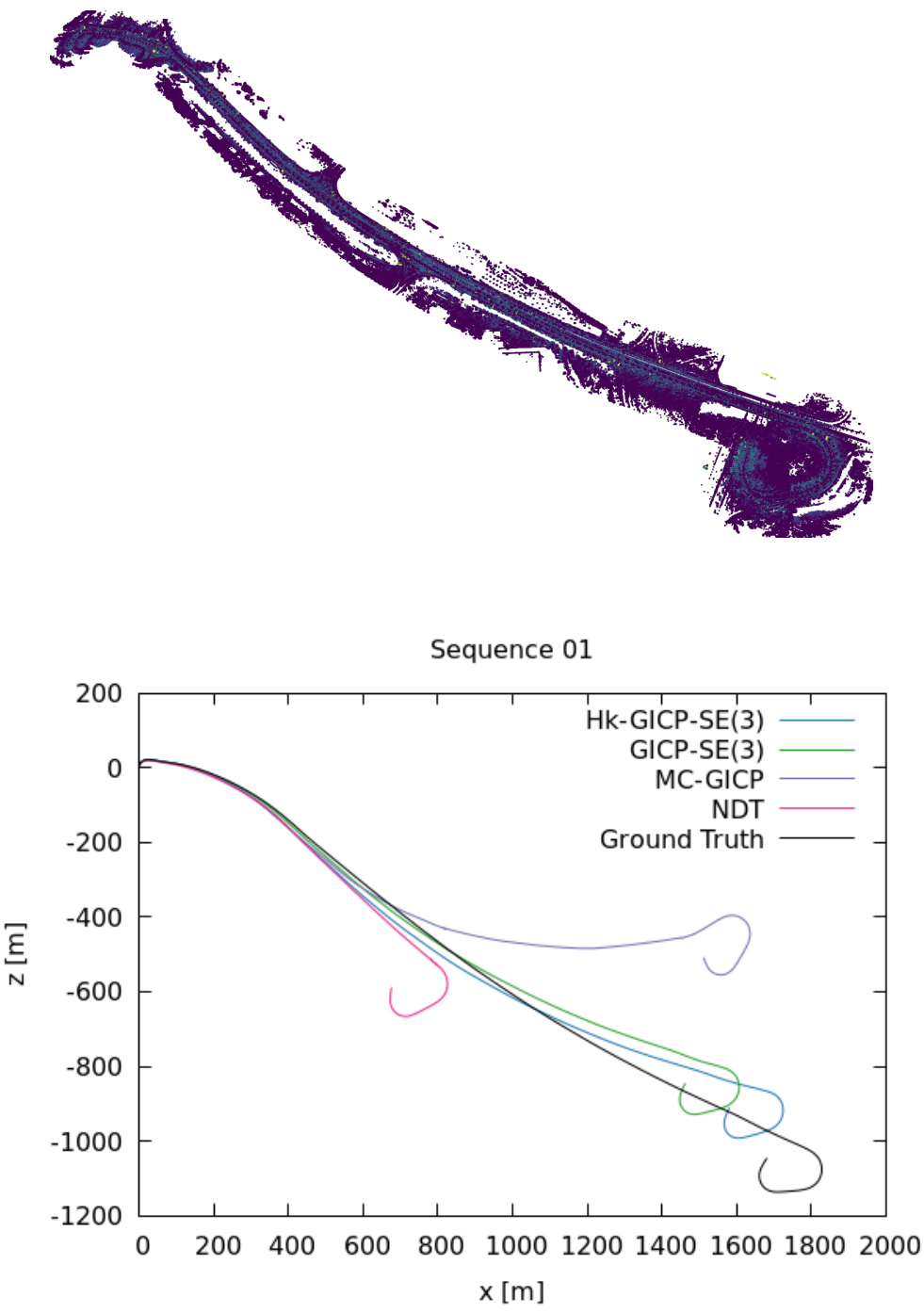



Figure 3.2: Results of the proposed method versus the benchmark algorithms on sequences 01 of the KITTI Odometry dataset. Above are the point clouds projected into the same reference frame using the estimated $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ odometry. This sequence is one of the more challenging ones, and we found that our proposed method had less transformation error by regularizing the cost on point cloud intensity.

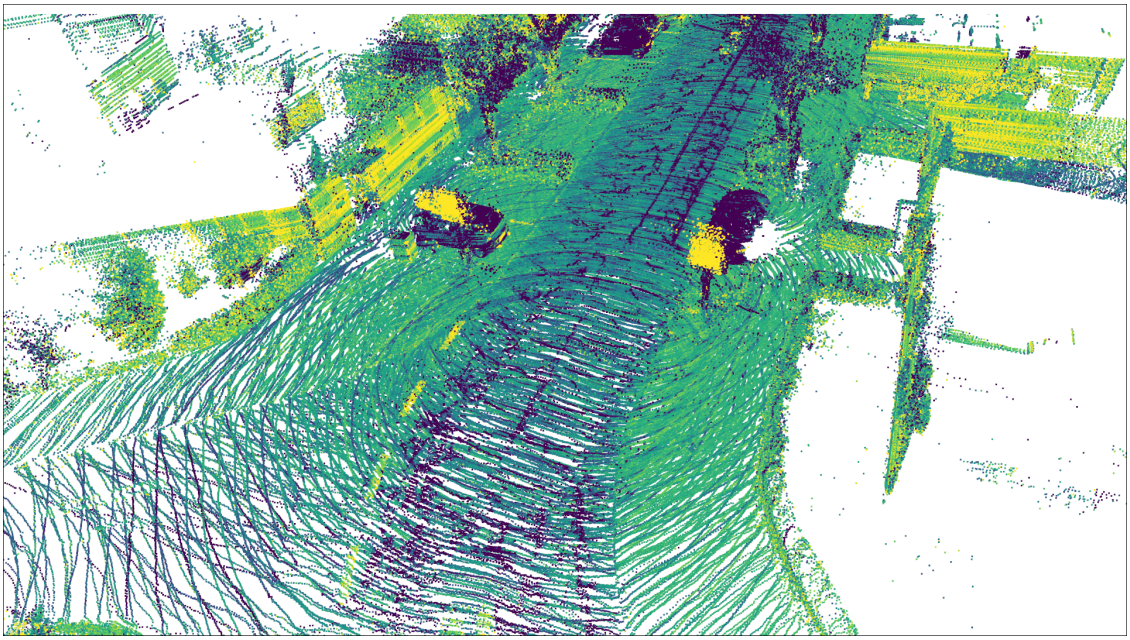


Figure 3.3: A detailed view of sequence 00 reconstructed using $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ odometry, labeled with intensity. Details such as lane lines and road signs are clearly visible, suggesting a good alignment.

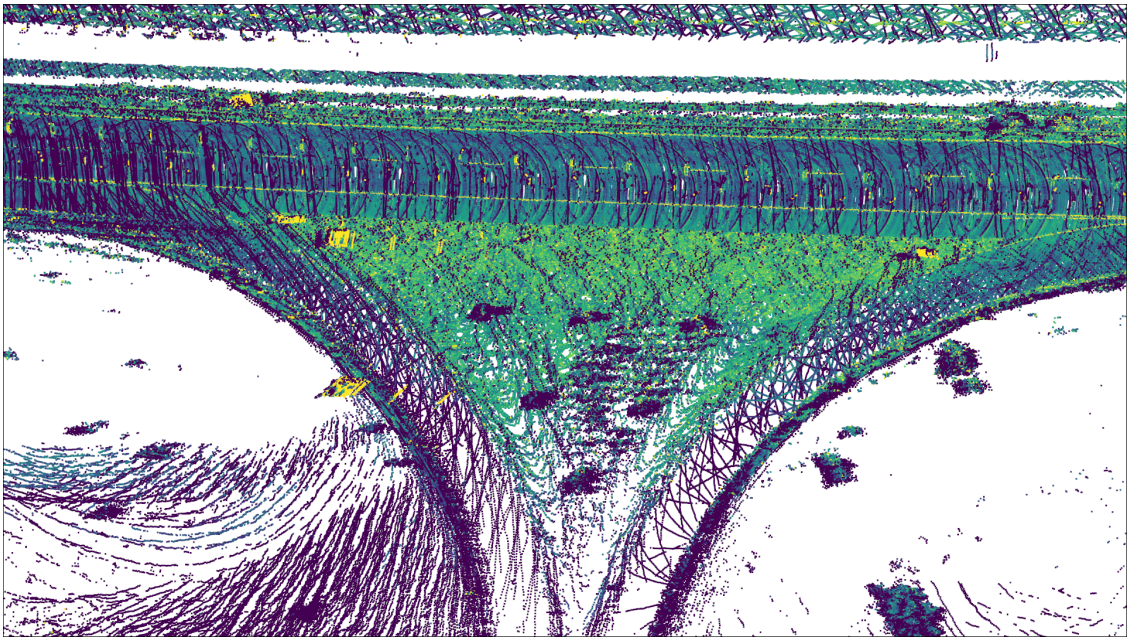


Figure 3.4: A detailed view of sequence 01 reconstructed using $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ odometry, labeled with intensity. Details such as lane lines and road signs are clearly visible, suggesting a good alignment.

Table 3.2: Results of the evaluation of $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ using the KITTI odometry benchmark as evaluated on the drift in translation, as a percentage (%), and rotation, in degrees per meter($^\circ/\mathrm{m}$). Best performances not including ties are in **bold**. Parameters were tuned on sequence 04 for both approaches, and so 04 is not included in the average.

| | | 00 | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | **Avg** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{H}_k$-GICP-SE(3) | t (%) | 1.96 | **7.26** | **2.39** | **1.43** | 2.61 | 1.74 | **1.29** | **1.40** | **1.98** | 2.15 | 2.63 | **2.28** |
| | r ($^\circ/\mathrm{m}$) | 0.0154 | 0.0203 | **0.0151** | 0.0230 | 0.0256 | 0.0148 | 0.0151 | **0.0173** | **0.0165** | 0.0162 | 0.0181 | **0.0160** |
| GICP-SE(3) | t (%) | 1.96 | 13.2 | 2.84 | 1.44 | **2.58** | 1.74 | 1.30 | 1.42 | 1.99 | **2.14** | 2.63 | 2.66 |
| | r ($^\circ/\mathrm{m}$) | 0.0154 | **0.0180** | 0.0179 | 0.0230 | 0.0254 | 0.0149 | 0.0151 | 0.0174 | 0.0166 | 0.0162 | 0.0182 | 0.0165 |
| MC-GICP | t (%) | 2.19 | 17.8 | 3.04 | 1.84 | 4.45 | 1.79 | 1.56 | 1.65 | 2.26 | 2.30 | 2.65 | 3.08 |
| | r ($^\circ/\mathrm{m}$) | 0.0174 | 0.0445 | 0.0164 | **0.0218** | **0.0246** | **0.0143** | 0.0159 | 0.0192 | 0.0175 | **0.0155** | **0.0170** | 0.0180 |
| NDT | t (%) | 1.69 | 49.94 | 3.38 | 3.07 | 1.59 | 2.54 | 0.90 | 2.06 | 4.29 | 2.57 | 3.97 | 5.07 |
| | r ($^\circ/\mathrm{m}$) | 0.0161 | 0.0333 | 0.0234 | 0.0262 | 0.0468 | 0.0271 | 0.0060 | 0.0311 | 0.0328 | 0.0207 | 0.0369 | 0.0249 |

GICP we re-implemented the algorithm following the description in the paper. Our implementation of MC-GICP is available with the provided code for this chapter along with parameters used for each algorithm. In the first experiment, LIDAR data is from KITTI odometry dataset Geiger, Lenz, and Urtasun (2012). In the second experiment, we use RGB-D data from the TUM RGB-D SLAM dataset Sturm et al. (2012) to generate point clouds where the intensity values are computed using RGB measurements. Table 3.1 lists the parameters used for our algorithm.

### 3.4.1 LIDAR: KITTI Odometry dataset

The KITTI benchmark provides evaluation metrics that compute error per distance traveled, or drift, as a percent for translation and $^\circ/\mathrm{m}$ in rotation. We used our proposed method to train functions, $f_s$ and $f_t$, on the intensity values provided by the LIDAR sensor. Parameters for all methods were tuned on sequence 04.

#### 3.4.1.1 Odometry Analysis

We first evaluated the proposed method versus our comparison methods in frame-to-frame odometry using the provided error metrics, an example is shown in Figure 3.2. Since all these methods use local gradient-based solvers, we seed the next frame with the solution of the previous, assuming there will not be a large change in velocity. The results for sequences 00 through 10 are presented in Table 3.2. We found that the proposed method performed better overall, with a translation drift of 2.27% versus 2.66% for the GICP-$\mathrm{SE}(3)$, 3.08% for MC-GICP, and 5.07% for NDT. For the sequences that GICP-$\mathrm{SE}(3)$ had good results, the proposed $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ performed similarly well without doing noticeably better. However, when GICP-$\mathrm{SE}(3)$ performed poorly, our method had noticeable improvement, suggesting the intensity regularizer contributes

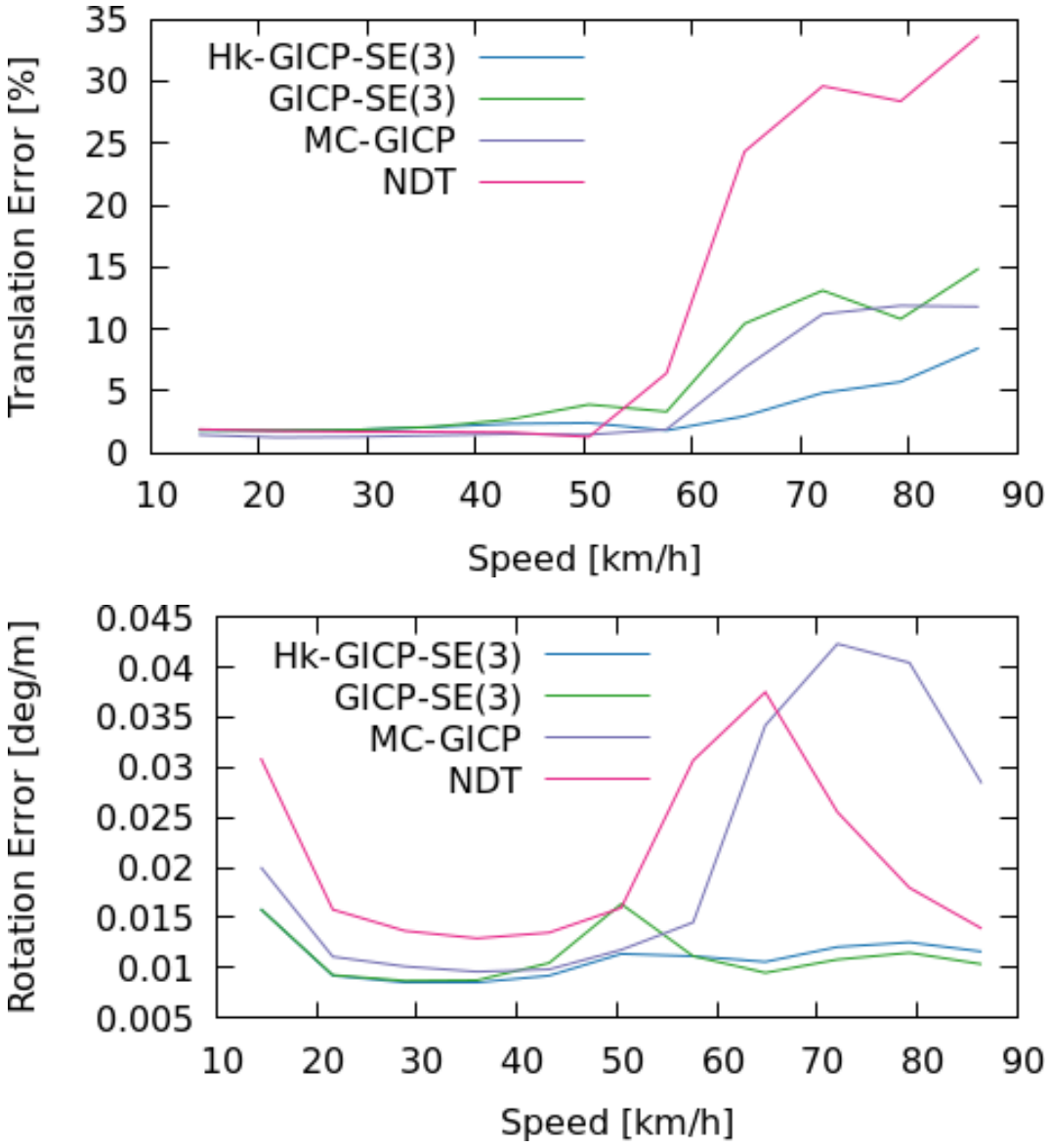



Figure 3.5: Average translation and rotation error vs speed on the KITTI Odometry dataset. We can see from the translation error that $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ has better results as the distance between point clouds increases.

to minimizing the effect of poor geometrical registration (Figure 3.4). All methods showed good performance in terms of the rotation error results, even though the proposed method performed slightly better on average, 0.0160 °/m versus 0.0165 °/m for the GICP-$\mathrm{SE}(3)$ approach.

Figure 3.5 shows the average translation and rotation errors versus speed. We observed that the proposed method performs better at higher speeds, suggesting the intensity regularization aided to expand the basin of convergence of the GICP algorithm. In addition, the proposed approach is competitive with many of the methods on the KITTI odometry leader board. Most of those methods are SLAM or filtering systems that take into account observations from multiple frames for each position estimate. It is a good indication that our frame-to-frame approach is already competitive with these methods, and leaves open the possibility of incorporating this registration approach into a SLAM system for future work.

#### 3.4.1.2 Convergence Analysis

We also used the KITTI Odometry dataset to evaluate the per-frame convergence of our proposed approach. To do so, we initialized the methods with the identity transformation and compared the initial error to the final error. The results of this analysis can be seen in Figure 3.6.

We can see that the proposed approach converges more consistently than both the purely geometric methods and MC-GICP which also incorporates the intensity information. MC-GICP only incorporates intensity information locally to each point, while our sparse model is a global approximation of the intensity, which in turn allows the regularizer to improve the convergence properties of the base algorithm.

We also analyzed the computation time each algorithm takes, shown in Figure 3.7. Since our method includes training the functions online, it does take longer than the compared methods. It is approximately four times slower than GICP-$\mathrm{SE}(3)$ and seven times slower then NDT. There are compromises that can be made when constructing the regularizer, such as fewer RVM training iterations, that would make our method approach GICP-$\mathrm{SE}(3)$ in terms of speed and performance. There is also a potential for parallelization in training the sparse model as well as the cost function evaluation. Particularly if we change from the sequential approach presented in Tipping, Faul et al. (2003) to the batch solution first derived in Tipping (2001). However, such approaches would come at the cost of runtime when only a single thread is available for computation. The many independent but identical operations in batch-RVM training and cost function evaluation make implementing a version for the fine-grained parallelism of a GPU attractive as future work. None of the methods evaluated were quick enough to operate at the update rate of the LIDAR sensor used in this dataset (10 $\mathrm{Hz}$). But the increased convergence performance of our approach suggest that it would work better when frames are dropped in an online system.

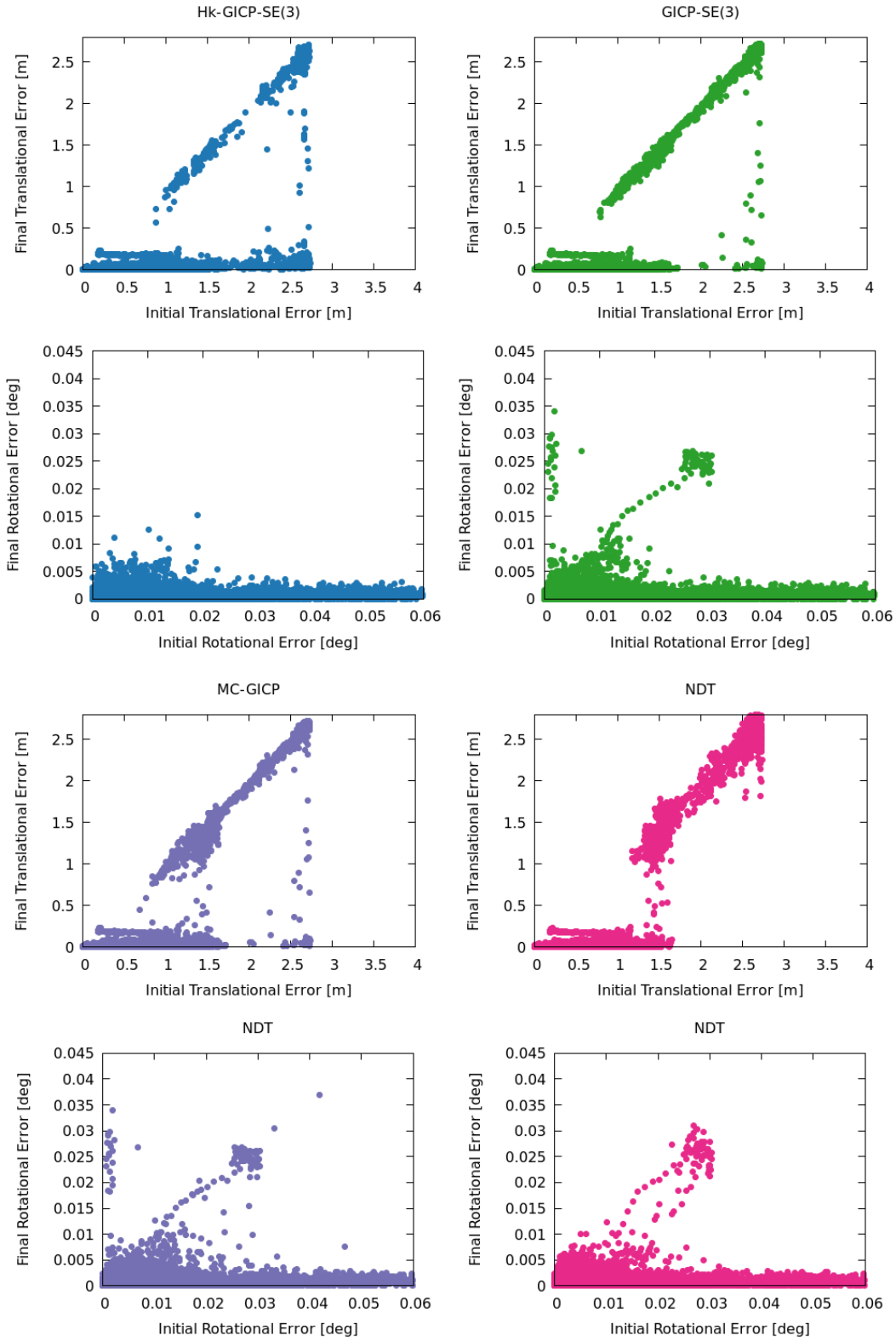



Figure 3.6: Scatter plots of initial error versus final error of the various methods on KITTI Odometry sequence 00 through 10. Our approach more consistently converged to the ground truth transformation.

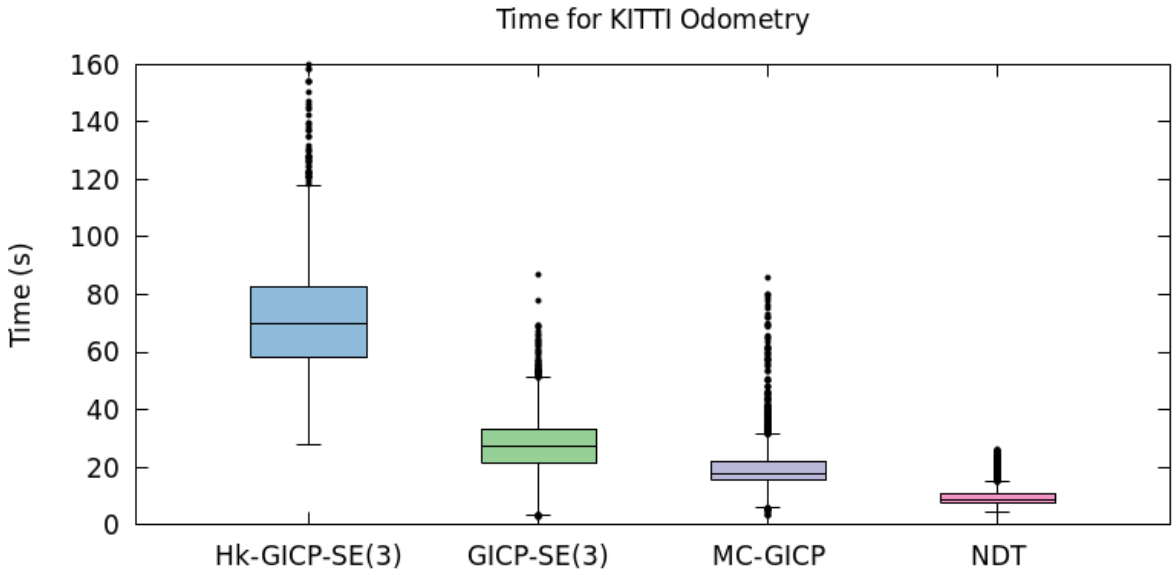



Figure 3.7: Timing comparison for the various algorithms on the KITTI Odometry dataset for the 23900 consecutive pairs in all sequences, 00 through 10.

### 3.4.2 RGB-D: TUM RGB-D SLAM dataset

The TUM RGB-D SLAM dataset Sturm et al. (2012) was collected indoors using a Microsoft Kinect and a motion capture system for ground truth trajectory. We used data from four sequences: Freiburg 1 desk, Freiburg 2 desk, Freiburg 3 no-structure-texture-far, and Freiburg 3 no-structure-texture-near-with-loop. Parameters for all methods were tuned on Freiburg 1 xyz and Freiburg 1 rph. Depth images were associated with RGB images using the provided python program. We trained the regularizer functions on the intensity obtained by averaging the RGB channels of the images. The results presented in Table 3.3 show the per frame drift of the five methods using the provided relative pose evaluation. Our approach does well in the two scenes with geometric structure (Freiburg 1 desk and Freiburg 2 desk) but show lower performance on the scenes that only have texture and no structure (both Freiburg 3 sequences). This presents the trade-off of the sparse Bayesian approach. The sparsity of support of the learned intensity functions, while suitable for convergence, performs poorly when there is no structure or local refinement. This is further illustrated in Figure 3.8 which provides sample images from one of the desk scenes and one of no structure scenes. It also includes CDF plots of translational error from the two desk scenes and from the two no structure scenes.

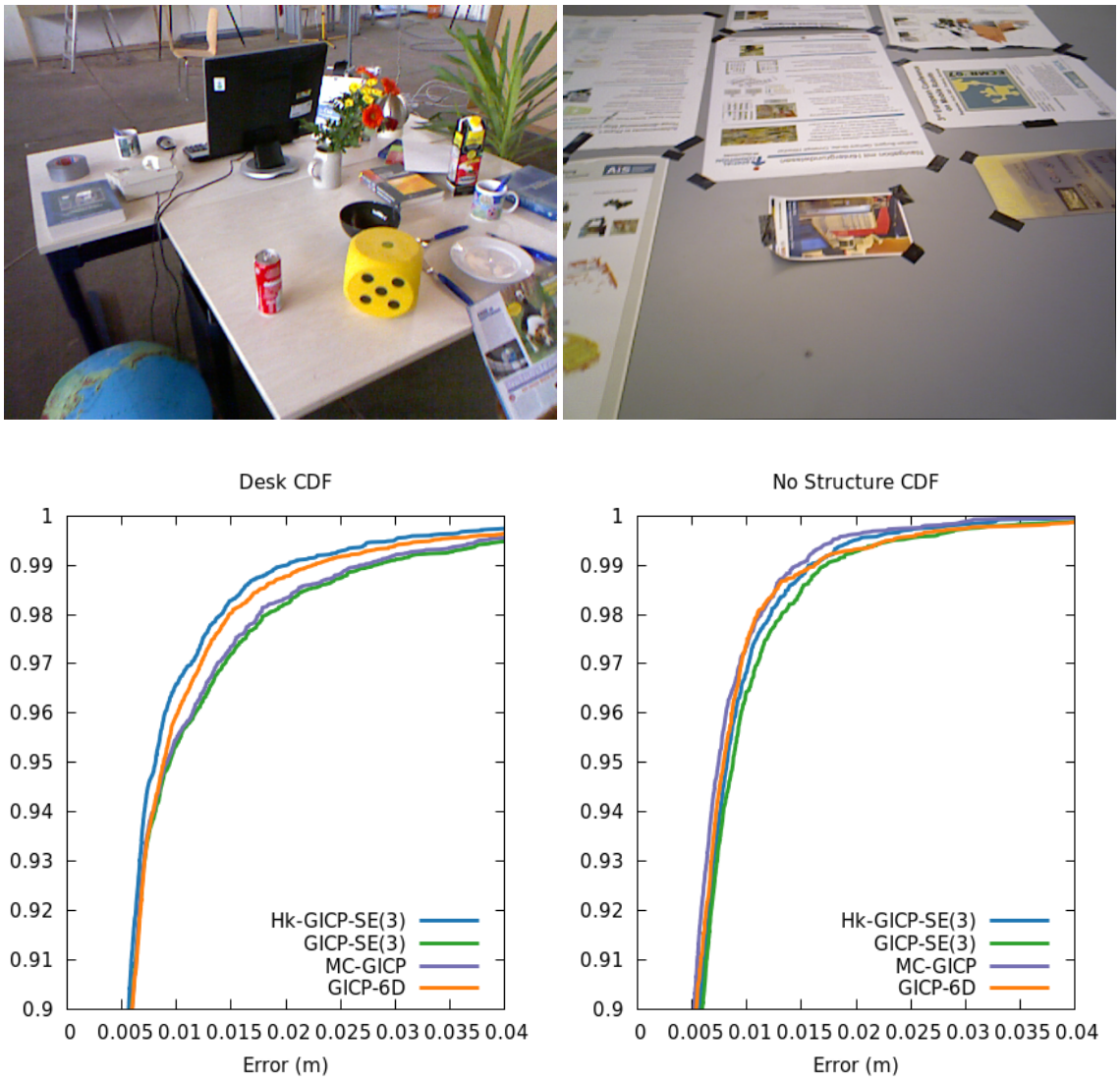



Figure 3.8: Cumulative distribution plots for translational error from the TUM RGB-D SLAM dataset, comparing Freiburg 1 desk and Freiburg 2 desk to the Freiburg 3 no structure sequences.

## 3.5 Conclusion

To reduce the effect of mis-associations in the registration problem, we presented an algorithmic approach to improve shape registration using regularizers represented in an RKHS. The RKHS are used to regress a function that represent an intensity function. The regularizer works by minimizing the discretely sampled distance between the target and source functions. We presented results on the KITTI Odometry and TUM RGB-D datasets using LIDAR and RGB-D sensors that showed promising improvements over relative transformation error when compared to related methods such as multi-channel GICP and GICP 6D.

Table 3.3: Evaluation of $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ on the TUM RGB-D SLAM dataset in per frame translation error in meters and rotation error in degrees. Parameters were tuned on separate sequences, Freiburg1 xyz and Freiburg1 rph.

| | | fr1/desk | fr2/desk | fr3/far | fr3/near |
|---|---|---|---|---|---|
| $\mathcal{H}_k$-GICP-$\mathrm{SE}(3)$ | mean t (m) | **0.00674** | **0.00137** | 0.00391 | 0.00328 |
| | median t (m) | **0.00517** | 0.00075 | 0.00330 | 0.00263 |
| | mean r (°) | **0.458** | **0.125** | 0.210 | 0.231 |
| | median r(°) | 0.360 | 0.068 | 0.181 | 0.197 |
| GICP-$\mathrm{SE}(3)$ | mean t (m) | 0.00775 | 0.00186 | 0.00390 | 0.00344 |
| | median t (m) | 0.00529 | 0.00075 | 0.00329 | 0.00265 |
| | mean r (°) | 0.519 | 0.144 | 0.211 | 0.237 |
| | median r(°) | 0.372 | 0.068 | 0.181 | 0.199 |
| MC-GICP | mean t (m) | 0.00747 | 0.00154 | **0.00371** | **0.00318** |
| | median t (m) | 0.00518 | 0.00075 | **0.00324** | 0.00263 |
| | mean r (°) | 0.492 | 0.131 | **0.202** | **0.229** |
| | median r(°) | 0.350 | **0.067** | **0.177** | **0.195** |
| GICP 6D | mean t (m) | 0.00677 | 0.00164 | 0.00390 | 0.00325 |
| | median t (m) | 0.00521 | 0.00075 | 0.00321 | 0.00263 |
| | mean r (°) | 0.461 | 0.134 | 0.217 | 0.235 |
| | median r(°) | 0.360 | 0.068 | 0.182 | 0.199 |
| NDT | mean t (m) | 0.02005 | 0.01074 | 0.01666 | 0.01665 |
| | median t (m) | 0.005546 | 0.000755 | 0.00345 | 0.002825 |
| | mean r (°) | 0.893 | 0.252 | 0.318 | 0.590 |
| | median r(°) | 0.405 | 0.071 | 0.217 | 0.229 |

There are areas to focus on when it comes to increasing the runtime performance. There could be a way of intelligently sampling points as basis vectors int he sequential training approach. There is also the potential to parallelize the algorithm on the GPU for increased performance. The proposed approach was less reliable in certain scenes of the TUM RGB-D dataset. When there was little geometric structure in the scene are regularizer did not do well as methods that directly incorporated color into the cost function. To improve in this aspect we could look at different kernels, or also adding color to the cost function, not just the regularizer.

In the future, we could incorporate the proposed approach into a SLAM or smoothing framework. A regularizer function could be trained on an area, which could be then used to localize into. The sequential training of the functions leads to direct ways of updating the map. We could also look at utilizing the probabilistic nature of the sparse Bayesian inference regularizer to make it more cohesive with the probabilistic motivation of GICP. And finally, regularizing to multi-

dimensional information channels could be a useful direction to explore, particularly RGB Ghaffari et al. (2019). We focused mostly on intensity in this work but curvature and normal information might also be useful. This method could also be used to incorporate deep feature vectors into the registration problem in a way that is not affected by data density issues or occlusions.

# CHAPTER 4

# 2D to 3D Line-Based Registration with Unknown Associations via Mixed-Integer Programming

In the previous two chapters we presented gradient-based optimization approaches to the sensor registration problem. Because our cost functions are non-linear, the solutions we find are only local minima. For mobile robotics, this means you need a reasonably accurate initial guess on the estimated transformation to successfully converge to the correct solution. In this chapter, we reformulate the registration problem so that it has a linear cost function and model the association variables as integer valued. Combining these we formulate sensor registration as a mixed-integer program (MIP), for which there are many off-the-shelf solvers capable of finding global solutions. In particular, in this work we focus on 2D-3D registration, a common framework that is needed for extrinsic calibration between camera and light detection and ranging (LIDAR) sensor, in which we may not have a good prior for.

## 4.1 Introduction

2D images and 3D data provide complementary representations of an environment; 3D data includes metric information, while 2D images report a rich visual representation. The rigid body transform between an image and 3D data must be accurately known in order to effectively perform geometric inference on their data. 2D to 3D registration is the problem that seeks to determine this transformation. Tasks that rely on accurate solutions to this problem include determining the extrinsic calibration between camera and LIDAR systems and localizing a camera into a 3D map. This problem is a subset of the larger registration problem, which estimates the transform between two inputs.

Fischler and Bolles (1981) proposed an early method for 2D to 3D registration dubbed the Perspective-n-Point (PnP) algorithm, which finds the perspective of a camera given a set of 2D point features and corresponding 3D points. Other early work also considered different line parameterizations within a geometric cost functions by Bartoli and Sturm (2001) and used within a


Figure 4.1: Alignment of 3D lines (green) with 2D lines (red) using the proposed approach.

bundle adjustment problem by Bartoli and Sturm (2005). More recently, there has been renewed interest in 2D to 3D line registration, described as the Perspective-n-Line (PnL), problem (Mirzaei and Roumeliotis (2011); Zhang et al. (2013); Přibyl, Zemčík, and Čadík (2016)).

There are two variables that are important to the registration problem: the rigid-body transformation and the set of associations between objects in the 2D data and objects in the 3D data. The associations are latent variables that play a large roll in most approaches to the registration problem. Many prior methods solve the problem for a set of known associations. In circumstances when a good prior on the transformation or associations is not available, this assumption can be problematic. There are methods that were developed to handle unknown associations. The algorithm random sample consensus (RANSAC) by Fischler and Bolles (1981) can be applied to these problems, by randomly sampling possible associations until enough inliers are found. Soft assign Pose from Orthography and Scaling with ITerations (SoftPOSIT) by David et al. (2004) was developed as a coordinate descent like approach the iteratively switches between finding the best associations and finding the best transformation. It was also extended to lines (David et al. (2003)), but this approach still relies on a well known prior for initialization. Bhat and Heikkilä (2014) perform a search over a set of uniformly sampled transformations. The search is used to prune potential line to line associations, after which they perform a final RANSAC procedure to refine the transformation.

Our proposed approach is to formulate the associations as binary variables in a linear problem, turning the 2D-3D registration problem into a MIP. Common approaches to generic MIP involve searching over the integer variables using the branch and bound algorithm. Others have explored developing bespoke branch and bound search algorithms over transformations for sensor registration. Yang, Li, and Jia (2013) bounded a point cloud in a sphere defined by a transformation cuboid to find the optimal point cloud registration. Parra Bustos, Chin, and Suter (2014) presented a similar approach, but used stereographic projections to bound the possible point locations. Campbell et al. (2017) alternated between bounding map points on rotation and transformation for camera based localization.

The naive way of performing a search is exhaustively evaluating all possibilities. Multiresolution search is related to branch and bound, but works by exhaustively searching a lower resolution of the data, to bound the desired resolution of the search. Using the fact that the translation component of an $\mathrm{SE}(3)$ transformation is usually in the reference frame of the sensor, bounds can be efficiently computed (Olson (2009); Wolcott and Eustice (2017b)).

Izatt, Dai, and Tedrake (2017) formulate the point cloud registration problem as a MIP by introducing a variable to represent what subregion of $\mathrm{SO}(3)$ they are currently evaluating in their branch and bound algorithm.

In this chapter, we present work on 2D-3D line-based registration. We propose several novel techniques including:

- a robust linear cost function,
- a registration algorithm for unknown associations formulated as a mixed-integer problem,
- evaluations on using the algorithm in localization applications.

We present an evaluation of the proposed algorithm on several real-world data sets in varying environments.

## 4.2 Problem Formulation

The registration problem involves finding the correct transformation, $\mathbf{T}_{AB} \in \mathrm{SE}(3)$, that transforms a set of 3D points in reference frame $B$ , $\{p_n^B\}, p_n^B \in \mathbb{R}^3$, into the reference frame $A$ of a set of 2D pixels, $\{i_m^A\}, i_m^A \in \mathbb{R}^2$, given a projection from 3D to 2D provided by camera intrinsic matrix $K$. In order to reduce the dimensionality of the problem, we look to use sparse set of line features. The projection operation

$$\widetilde{i}_j^A = \pi\left(p_j^A\right) = Kp^A \tag{4.1}$$

is line preserving. This means that any three collinear points in $\mathbb{R}^3$ are also collinear after the projection operator $\pi(\cdot)$. PnL algorithms exploits this fact to perform registration.

### 4.2.1 Plücker Coordinates

There are various ways to parameterize lines in 2D and lines in 3D. We follow the approach presented by Bartoli and Sturm (2001) where lines are represented as Plücker Coordinates. If $p_s$ is a point in $\mathbb{R}^3$ that represents the start of a 3D line segment and $p_e$ is the end, the corresponding Plücker Coordinates can be computed as follows

$$L = \begin{bmatrix} p_e \times p_s \\ p_e - p_s \end{bmatrix} \tag{4.2}$$

where $p_e \times p_s$ represents the normal of the line and $p_e - p_s$ is the direction of the line. To transform Plücker coordinates, the following $6 \times 6$ matrix is used

$$\mathcal{T}_{AB} = \begin{bmatrix} R_{AB} & [t_{AB}]_x R_{AB} \\ 0 & R_{AB} \end{bmatrix}$$

where $[\cdot]_x$ represents the operation of turning a translation in $\mathbb{R}^3$ into a skew symmetric matrix.

### 4.2.2 Line-Based Registration

One of the advantages of Plücker parametrization of 3D lines is the operation that represents a rigid body transformation of a line is linear with respect to the transformation parameter $\mathcal{T}_{AB}$. With the only non-linearity contained within the constrains of the $\mathrm{SE}(3)$ group it is derived from. To further utilize that advantage, we propose to modify the cost function presented by Bartoli and Sturm (2001), which maintains linearity with respect to the transformation

$$d\left(\{i_e, i_s\}_m^A, L_n^B\right) = |i_e \cdot \hat{L}_n^A| + |i_s \cdot \hat{L}_n^A|, \tag{4.3}$$

in which $\hat{L}_n^A$ is the normal of the line in the image reference frame

$$\hat{L}_n^A = \det(K) K^{-\top} \mathcal{P}_{AB} L_n^B$$

and $\mathcal{P}_{AB}$ is the subset of the transformation that affects the normal

$$\mathcal{P}_{AB} = \begin{bmatrix} R_{AB} & [t_{AB}]_x R_{AB} \end{bmatrix}. \tag{4.4}$$

This is the cost function presented by Bartoli and Sturm (2001) with the exception of taking the absolute value of the dot product instead of the second power. Effectively the difference between the minimizing the $\ell_1$ versus the $\ell_2$ norm. The problem can now be expressed

$$\underset{\mathcal{P}_{\mathcal{AB}}}{\arg\min} \sum_{\{n,m\}\in\mathcal{I}} d\left(\{i_e, i_s\}_m^A, L_n^B\right), \tag{4.5}$$

which is linear except for the constraints within $\mathcal{P}_{AB}$ to make it a valid transformation.

The distance measure (4.3) is the $\ell_1$ version of the formulation used in other applications, including stereo mapping Zhang et al. (2015), stereo visual-inertial odometry He et al. (2018), stereo simultaneous localization and mapping (SLAM) Gomez-Ojeda et al. (2019) and monocular SLAM Zuo et al. (2017). One notable difference is that the dot products are normalized by the size of $\hat{L}_n^A$,

$$\frac{(i_e \cdot \hat{L}_n^A)^2}{||\hat{L}_n^A||} + \frac{(i_s \cdot \hat{L}_n^A)^2}{||\hat{L}_n^A||}, \tag{4.6}$$

which can not be done while keeping the problem linear. Instead we normalize $L_n^B$ when it is first computed. Přibyl, Zemčík, and Čadík (2016) used a different metric based on Plücker coordinates that minimizes the angle between the norm of 2D lines to the corresponding normal of the 3D line projected into the image reference frame.

Most approaches to the 2D to 3D registration problem assume a known set of associations $\mathcal{I}$. Even if the set is not provided, if a good initial guess for the projection transformation $\mathcal{P}$ is known, a nearest neighbor heuristic could be used. We seek to minimize (4.5) when no prior for $\mathcal{P}$ or $\mathcal{I}$ are available.

There are good reasons associations are usually treated as latent variables and not solved for. The complexity of the set of possible associations grows quickly over the size of the data. At best, if the problem is formulated such that there are no outliers and each member can only be associated with one of the opposite set, the complexity is factorial. If there can be many-to-one associations both directions, the complexity worsens to $n^n$. On top of that, allowing for outlier data adds even more potential associations. This is shown with a log-scale in Figure 4.2. Luckily, the space of possible associations generally does not need to be exhaustively searched. If we formulate our approach as a mixed-integer problem where the association variable is integer valued, we can leverage the efficiencies available in modern generic solvers. Most mixed-integer solvers use a branch and bound approach to avoid searching all possible integer values, and have built in heuristics to further reduce the time it takes to solve these problems.

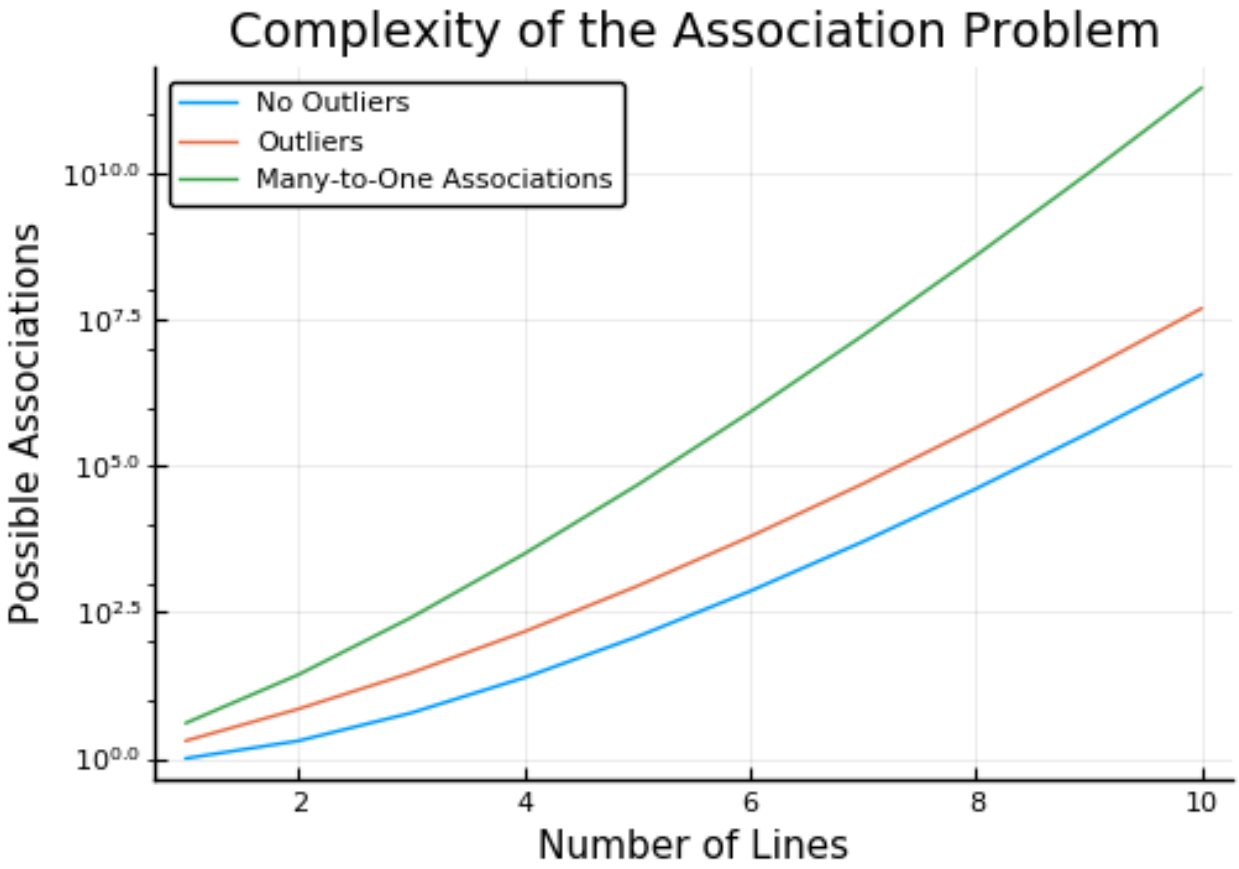



Figure 4.2: The number of possible associations (log-scale) versus the number of 2D and 3D lines for various types of associations.

## 4.3 Method

One advantage of the formulation in Equation (4.5) is that the residual is linear with respect to the projection parameter. Přibyl, Zemčík, and Čadík (2016) and Bartoli and Sturm (2001) solve similar problems by assembling a measurement matrix $M$ and then find the null space of that matrix by performing an SVD decomposition. Their solutions do not satisfy the constrains of the $\mathrm{SE}(3)$ group, but they show they can be fitted into the group to find accurate solutions.

Our approach is to minimize the $\ell_1$ norm of Equation (4.5). The $\ell_1$ norm is more robust to outlier and allow us to use off-the-shelf linear program solvers.

### 4.3.1 Mixed-Integer Formulation

To simultaneously solve for the transform $\mathcal{P}_{AB}$ and the association set $\mathcal{I}$, we introduce a binary variable $s_{mn}$ that indicates if lines $L_n^B$ and $\{i_e, i_s\}_m^A$ are not associated (so it is 0 if they are associated). This association variable controls if a large $\lambda$ constant is subtracted from the distance, making the causing it to not affect it's corresponding positive-valued slack variable $\alpha_{mn}$. With that we get the following

$$
\begin{aligned}
\min_{p,s,\alpha} \quad & \sum_n \sum_m \alpha_{nm} \\
\text{s.t.} \quad & d\left(\{i_e, i_s\}_m^A, L_n^B\right) - \lambda s_{nm} < \alpha_{nm} \\
& d\left(\{i_e, i_s\}_m^A, L_n^B\right) + \lambda s_{nm} > -\alpha_{nm} \\
& \sum_{j=0}^{M} s_{jn} = M - 1 \\
& \sum_{j=0}^{N} s_{mj} = N - 1 \\
& \alpha_{nm} \geq 0 \\
& s_{nm} \in \{0, 1\}
\end{aligned}
\tag{4.7}
$$

which is a mixed-integer linear problem.

#### 4.3.1.1 Handling Outliers

The formulation presented in Equation (4.7) allows for an easy extension to handle outliers. The equality on the binary variable gets changed to the inequalities

$$
\sum_{j=0}^{M} s_{jn} \geq M - 1
$$

$$\sum_{j=0}^{N} s_{mj} \geq N - 1$$

with similar changes to the summation over the other dimension of $s$.

#### 4.3.1.2 Field of View Constraint

Because of the setup of our problem there are other constraints we can leverage to reduce runtime and increase accuracy. One is that if there is a match for a 3D line $L_m$, some part of it must have been projected into the image plane. This can be approximated with the following equations.

$$\frac{u_{\max} - c_u}{f_u} \left(R_3 \cdot p + t_{\max}\right) \geq \left(R_1 \cdot p + t_{\max}\right)$$

$$\frac{v_{\max} - c_v}{f_v} \left(R_3 \cdot p + t_{\max}\right) \geq \left(R_2 \cdot p + t_{\max}\right)$$

### 4.3.2 Constraining the Problem to $\mathrm{SE}(2)$

For some applications, we would like to estimate a transformation in just $\mathrm{SE}(2)$. This reduces the dimensionality of the problem and speeds up the process of finding a solution. Without heavily changing our problem, we can add constraints to a transformation matrix $\mathcal{P}$ that keep the solution in $\mathrm{SE}(2)$.

In many camera reference frames, the $y$ axis points up. If we would like to estimate the flat movement of the camera then we would want to constrain our problem to translations along the $x$ and $z$ axis, and rotations around $y$ axis. Our transformation from Equation (4.4) becomes

$$\begin{aligned} \mathcal{P} &= \left[ \begin{bmatrix} c_y & 0 & -s_y \\ 0 & 1 & 0 \\ s_y & 0 & c_y \end{bmatrix} \begin{bmatrix} 0 & -z & 0 \\ z & 0 & -x \\ 0 & x & 0 \end{bmatrix} \begin{bmatrix} c_y & 0 & -s_y \\ 0 & 1 & 0 \\ s_y & 0 & c_y \end{bmatrix} \right] \\ &= \begin{bmatrix} c_y & 0 & -s_y & 0 & -z & 0 \\ 0 & 1 & 0 & c_y z - s_y & 0 & -s_y z - c_y x \\ s_y & 0 & c_y & 0 & x & 0 \end{bmatrix} \end{aligned} \tag{4.8}$$

and we therefore only have 8 elements of the $3 \times 6$ matrix we need to estimate.

### 4.3.3 Fitting to a Valid Transformation

To fit estimated transformation $\mathcal{P}_{AB}$ so that it satisfies the constraints of the $\mathrm{SE}(3)$ group, we first take the SVD of the left sub-block of the our estimated parameter, or $U\Sigma V^\top = \mathcal{P}_{AB}^{[1:3,1:3]}$, then

$$R = UV^\top$$

gives us the closest orthogonal matrix. If the $\det\left(UV^\top\right)$ is $-1$ (making it a reflection instead of a rotation) we can multiply the third column of $U$ by $-1$ to make it a proper rotation. We can find the skew symmetric matrix via

$$[t]_x = \frac{1}{2}\left(\mathcal{P}_{AB}^{[1:3,4:6]}R^\top - \left(\mathcal{P}_{AB}^{[1:3,4:6]}R^\top\right)^\top\right)$$

which is the closest skew symmetric matrix to $\mathcal{P}_{AB}^{[1:3,4:6]}R^\top$ in the sense of the Frobenius norm. It can be further decomposed to recover the translation component of the transformation. When we constrain the problem to $\mathrm{SE}(2)$ we still follow this procedure. Our constraints described in Subsection 4.3.2 keep our solution in the desired form, even after this fitting procedure, and we can ignore the $y$ component and rotations around $x$ and $z$.

## 4.4 Evaluation

We evaluated our approach on two datasets, the publicly available Oxford VGG Multiview Dataset[1], and a dataset collected with an autonomous vehicle platform. We implemented our approach using the Julia language and the JuMP (Dunning, Huchette, and Lubin (2017)) optimization framework which offers an interface to a variety of mixed-integer solvers. We used the Gurobi linear program and mixed-integer program solvers. We implemented the comparison methods in Julia as well to make timing results comparable. Timing results that are presented are from a computer with dual-socket Intel Xeon ES-2690 cpus and 128 GB of RAM.

Many formulations of the PnL problem using Plücker coordinates use the $\ell_2$ version of the cost function shown in Equation 4.6. For comparison we evaluate with respect to another version presented by Přibyl, Zemčík, and Čadík (2016), whose residual is also linear with respect to $\mathcal{P}_{AB}$.

### 4.4.1 VGG Multiview Dataset

The VGG Multiview Dataset provides several sequences of images of various scenes around the campus of the University of Oxford. As well as the images, they include pre-extracted 2D lines

[1] http://www.robots.ox.ac.uk/ vgg/data/data-mview.html

Figure 4.3: Comparison between the SVD approach presented by Přibyl et al. and our linear $\ell_1$ approach on the VGG Corridor sequence with known associations. Red lines are 2D lines from the image while green lines are 3D lines. From the the Corridor sequence.

from the images, and a set of 3D lines constructed through bundle adjustment from all the images, as well labeled associations between the 2D and 3D lines. This allows us to test our approach agnostic to the line extractor used.

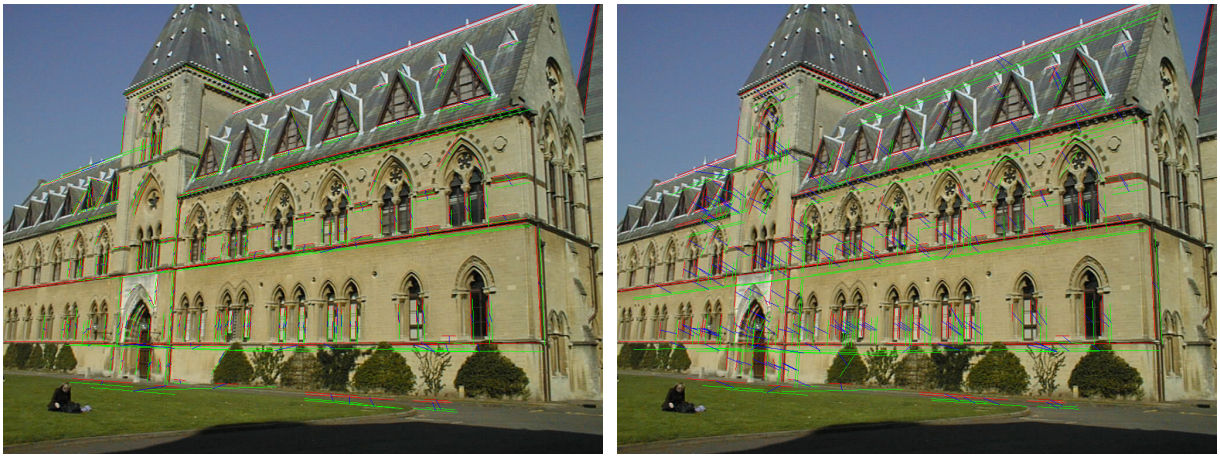
We first evaluated how our $\ell_1$ cost function with known associations performed against Přibyl et al. SVD solution. The results can be seen in Table 4.1. Our method performs similarly, though with slight improvements. Figure 4.3 gives some qualitative examples from each sequence. In general, both methods perform similarly, with our $\ell_1$ approach doing slightly better in the corridor scene (top) while the approach of Přibyl et al. performing better on the library scene (bottom).

We also evaluated our approach to simultaneously solving for transformation in $\mathrm{SE}(2)$ and for the associations. In this evaluation we subsampled a random permutation of the lines to get a distribution of errors versus the number of lines. We also modified the transformation between the 3D and 2D data to be only in $\mathrm{SE}(2)$. The result are shown in Figure 4.8, where we compare to our linear method and the cost function of Přibyl et al. Only the $\mathrm{SE}(2)$ method was constrained to find the solution in a lower state-space, and the error for all 3 methods was computed in $\mathbb{R}^3$ and $\mathrm{SO}(3)$.

These results show that our method, constrained to $\mathrm{SE}(2)$ finds the correct transformations, though at the cost of runtime complexity. It performs as well as state of the art approaches that require associations. Currently our system take prohibitively long for scenes that have more then $20 \times 20$ lines in 3D and 2D with unknown associations. This does limit what environments our

Figure 4.4: Comparison between the SVD approach presented by Přibyl et al. and our linear $\ell_1$ approach on the VGG Corridor sequence with known associations. Red lines are 2D lines from the image while green lines are 3D lines. From the Merton College I sequence.

Figure 4.5: Comparison between the SVD approach presented by Přibyl et al. and our linear $\ell_1$ approach on the VGG Corridor sequence with known associations. Red lines are 2D lines from the image while green lines are 3D lines. From the Wadham College sequence



(a) Přibyl et al. (b) $\ell_1$

Figure 4.6: Comparison between the SVD approach presented by Přibyl et al. and our linear $\ell_1$ approach on the VGG Corridor sequence with known associations. Red lines are 2D lines from the image while green lines are 3D lines. From the University Library sequence.

| | Přibyl et al. | | $\ell_1$ | |
|---|---|---|---|---|
| **Sequence** | **Translation (m)** | **Rotation (°)** | **Translation (m)** | **Rotation (°)** |
| Corridor | 0.119 | 0.33 | 0.055 | 0.14 |
| Melton I | 0.501 | 0.76 | 0.104 | 0.09 |
| Melton II | 0.316 | 0.34 | 0.174 | 0.09 |
| Melton III | 0.282 | 0.42 | 0.198 | 0.15 |
| Wadham | 0.579 | 0.77 | 0.530 | 0.51 |
| Library | 1.154 | 1.47 | 1.036 | 1.67 |

Table 4.1: Results on the VGG Multiview dataset with known associations, comparing the average error of the estimated transformation by the method by Přibyl et al. with our proposed $\ell_1$ approach.

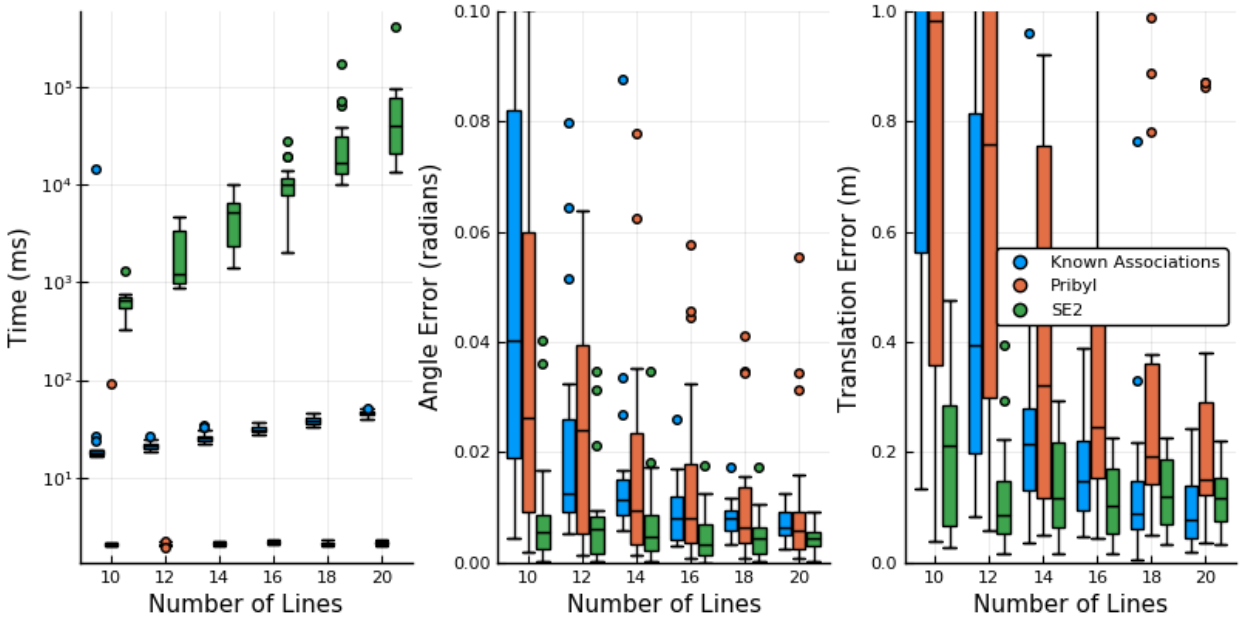



Figure 4.7: VGG Results comparing our method with known associations, Přibyl et al. with known associations, and our method constrained to $\mathrm{SE}(2)$ with out known associations. We can see that with our proposed method performs well even without being provided associations.

approach can be run in, but further preprocessing could eliminate lines with low probabilities of having a match. Finding a full $\mathrm{SE}(3)$ solution also add to the complexity of the problem. There are many tasks were $\mathrm{SE}(2)$ is sufficient, such as that of our next section.

### 4.4.2 Autonomous Vehicle Dataset

We also evaluated our proposed approach on data collected with an autonomous vehicle. This evaluation was designed to test how our approach would do in an application where we wanted to localize into a prior map of line features.

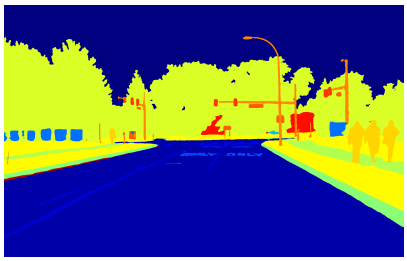
The data used in this evaluation was collected in two stages. In the first stage the vehicle drove the route and data from GPS, IMU, LIDAR, and cameras was combined in an offline SLAM system to generate a map of 3D line features. In the second stage the vehicle drove the route again to gather 2D line feature detections to use in the evaluation. We constrained the estimate of the transformation using the approach we presented in 4.3.2.

Figure 4.9 shows the process of extracting 2D Line features from the data. We first take the classes provided by a semantic segmentation network. We then keep any pixel whose most likely label corresponds to linear features such as road lines, curbs, and poles. Individual instances are then extracted with a 4-neighbor connected component algorithm. We perform a SVD on the pixel indices of each component to get the direction of the line. This is done by creating a 2

Figure 4.8: Results form the VGG dataset with unknown associations. Green represents 3D lines, red 2D lines, and blue connects the pairs association by our mixed-integer formulation.

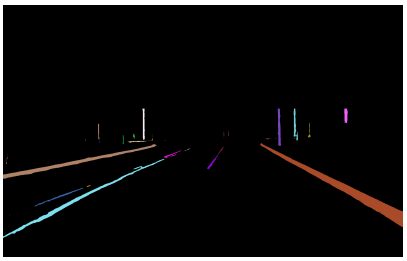
by 2 correlation matrix of the $u$ and $v$ pixel indices of each instance. SVD is performed on the correlation matrix, and the direction of the line, in image coordinates is the left singular vector corresponding to the largest singular value. We then find the max and min of the dot product between the direction of the line, and a line going from the mean of the component to each pixel coordinate. That gives us the extent and therefore the endpoints of the line.

We evaluated our approach on multiple frames of the driving data with unknown associations.



(a) Semantic Labels

(b) Linear Features



(c) CC Segements


(d) 2D Lines plotted over the original image

Figure 4.9: Illustration of the line extraction procedure used for the driving data. We first take the semantic label of the image (a) and extract the classes that correspond to linear features (b) (road paint, curbs, and poles). We then compute the connected components of those labels (c). Finally we compute the covariance matrix of the pixel coordinates of each segment and perform a SVD decomposition to determine the direction of the line(d). This approach consistently gave us good line features for use in the 2D to 3D registration process.

| Translation (m) | Rotation (°) | Runtime (s) |
|---|---|---|
| 0.23 | 5.43 | 1.691 |

Table 4.2: Results on the driving dataset with unknown associations. Error statistics are comparable to what was shown on the VGG Mulitview dataset. The average is affected mostly by a frame that settled on the wrong associations.

We did it to simulate a localization task. We used the 3D line map as a prior and used the image to find the $\mathrm{SE}(2)$ transformation that would localize the camera into the map. The qualitative results can be seen in Figure 4.12 and the average error and runtime are in Table 4.2. For most of the frames our approach performed well, finding the correct associations and accurate transformations. Figure 4.11 shows a case where it found incorrect associations and was pulled off the proper transformation. Given that our approach finds the global minimum of our cost function, this suggest that this error is a shortcoming of our line parameterization and distance functions. Even with that mis-association, the error was only 14°and 0.41m. The average error suggests the system could be used to localize vehicle when no prior is available, and be able to know which lane and which direction the vehicle is traveling in. Overall the system performed well given the noisy nature of the real world data. The average runtime of around 1 second is slower the data rate of the camera, but is fast enough to used online. It is conceivable to see this being a part of a system that performs the initial localization of the platform into a prior map, and is rerun whenever a significant divergence from the map is detected.

## 4.5 Conclusions

Many applications in robotics require estimates of parameters without informative priors, both in terms of transformation and data association. We propose an approach to 2D-3D registration that requires no prior on the transformation or the data associations. We did so by formulating it as a mixed-integer program where the associations were represented as binary variables. In addition, we showed a modification of our approach that can accomindate outliers, a way of constraining the registration to $\mathrm{SE}(2)$, and a way to add a constraint that keeps the 3D lines in the image plane of the camera. We evaluated our approach our approach on the VGG dataset and showed that with known association it performs well as the state of the art method presented by Přibyle et al. We also showed that our approach works without known association, unlike other current approaches. We also presented results from an autonomous vehicle dataset. The evaluations showed the approach could be useful for localizing into a map of 3D line features using data collected from a camera, with acceptable accuracy and runtime results.

Figure 4.10: Data collected from an autonomous vehicle platform. This shows the estimate of our approach for three frames. Green is a line in 3D, red is a line detected in 2D, and blue shows an estimated association.

Computational complexity remains a challenge, and limits the rate we are able to run our approach. Investigating methods to improve runtime, such as better constraints and preprocessing of the data, would be useful future work. Also, efficient extensions to $\mathrm{SE}(3)$ is important for a broader application of the approach. As is the use of relaxations on the nonlinear constrains of the $\mathrm{SO}(3)$ group, such as those presented in Dai, Izatt, and Tedrake and Izatt, Dai, and Tedrake (2017) in which they replace the $\mathrm{SO}(3)$ constraint with a piece-wise linear constraint. Further developing the approach into a full localization system is also promising future work.


Figure 4.11: Data collected from an autonomous vehicle platform. This shows the estimate of our approach for three frames. Green is a line in 3D, red is a line detected in 2D, and blue shows an estimated association.

Figure 4.12: Data collected from an autonomous vehicle platform. This shows the estimate of our approach for three frames. Green is a line in 3D, red is a line detected in 2D, and blue shows an estimated association. This frame estimated an incorrect association and transformation, while the other two converged to the correct results.

# CHAPTER 5

# Conclusion

In this thesis we presented work pushing the state of the art in various sensor registration tasks. We presented new ways to combine semantic and geometric objectives, effectively use transformation invariant information, and jointly optimize data association and geometric residuals. We evaluated our work on publicly available datasets for both accuracy and convergence properties, and compared our results to other recent work, and showed improved performance. The goal of this thesis was to leverage advances in machine learning, artificial intelligence, and optimization to improve one of the fundamental tasks in state estimation for mobile robotics. That being sensor registration.

## 5.1 Contributions

In this thesis we presented three contributions:

- We presented an algorithm for joint semantic-geometric point cloud registration that uses the expectation-maximization (EM) approach. In this approach semantic classification results from a CNN inform the data association in the registration problem. Not only does this approach minimize geometric residuals, it also maximizes semantic consistency, increasing the reliability of down stream joint semantic and geometric tasks. This work is discussed in Chapter 2.

- We presented work to refine sensor registration using viewpoint invariant features. The method in Chapter 2 is a local, gradient-based method similar to other iterative closest point (ICP) algorithms. We utilize viewpoint invariant features, namely intensity, for better alignments and convergence. We do this by training a sparse Bayesian representation of the invariant feature and use the distance between regressed representation as a regularizer on the registration problem. This work is discussed in Chapter 3.

- We presented work to further improve sensor registration by reformulating it as a mixed-integer program (MIP) which jointly solves for the data association and the rigid body transformation. This approach allows us to solve the registration problem without strong priors on the transformation or data associations. We do this for a two-dimensional (2D) to three-dimensional (3D) registration problem, which can represent localizing a camera into a 3D map, or determining the extrinsic calibration between a camera and LIDAR sensor. We evaluate this approach using linear semantic features such as poles, curbs, and road paint. This contribution is discussed in Chapter 4.

## 5.2 Future Work

Since point cloud registration was first studied forty years ago, sensor and compute hardware have advanced significantly. On top of that, other areas of robotic perception have experienced advances in algorithmic development. The goal of this thesis was to present methods that align these contemporary approaches to perception with the classic task of sensor registration.Towards this, this thesis explored many areas relevant to modern uses of sensor registration for mobile robotics. However, this thesis is not an exhaustive evaluation for all potential areas of research in registration. Moreover, the work presented in this thesis has itself opened up new paths for future research. This section highlights some of those areas, and presents initial results for a few of them.

### 5.2.1 GPU Accelerated Algorithms

While we presented methods that pushed the state of the art in the accuracy of sensor registration, one unwanted and unintended theme of this thesis is that most of the proposed approaches are slower than the data rate of the sensors they use as input. This is expected when developing new algorithms but to be practical we need to look into ways of speeding up their runtime. One approach is to leverage the massive parallelism of modern GPUs. Over the last decade, there has been a 40 times increase in the number of stream processors in a top-line Nvidia GPU, as shown in Figure 5.1. A GPU is not as capable as a central processing unit (CPU) on individual computations, but has the ability to do the same work on many threads at the same time. The paradigm that GPUs operate under is single input, multiple data (SIMD), which means they run the same computation on many data points, synchronously. This is less flexible then multi-threading on a CPU, but the number of stream processors (which are what perform the GPU computations on the data) on modern GPUs far exceed the number of cores on a modern CPU. SIMD also fits the way many registration algorithms work, where they run the same computation on many points at the same time.

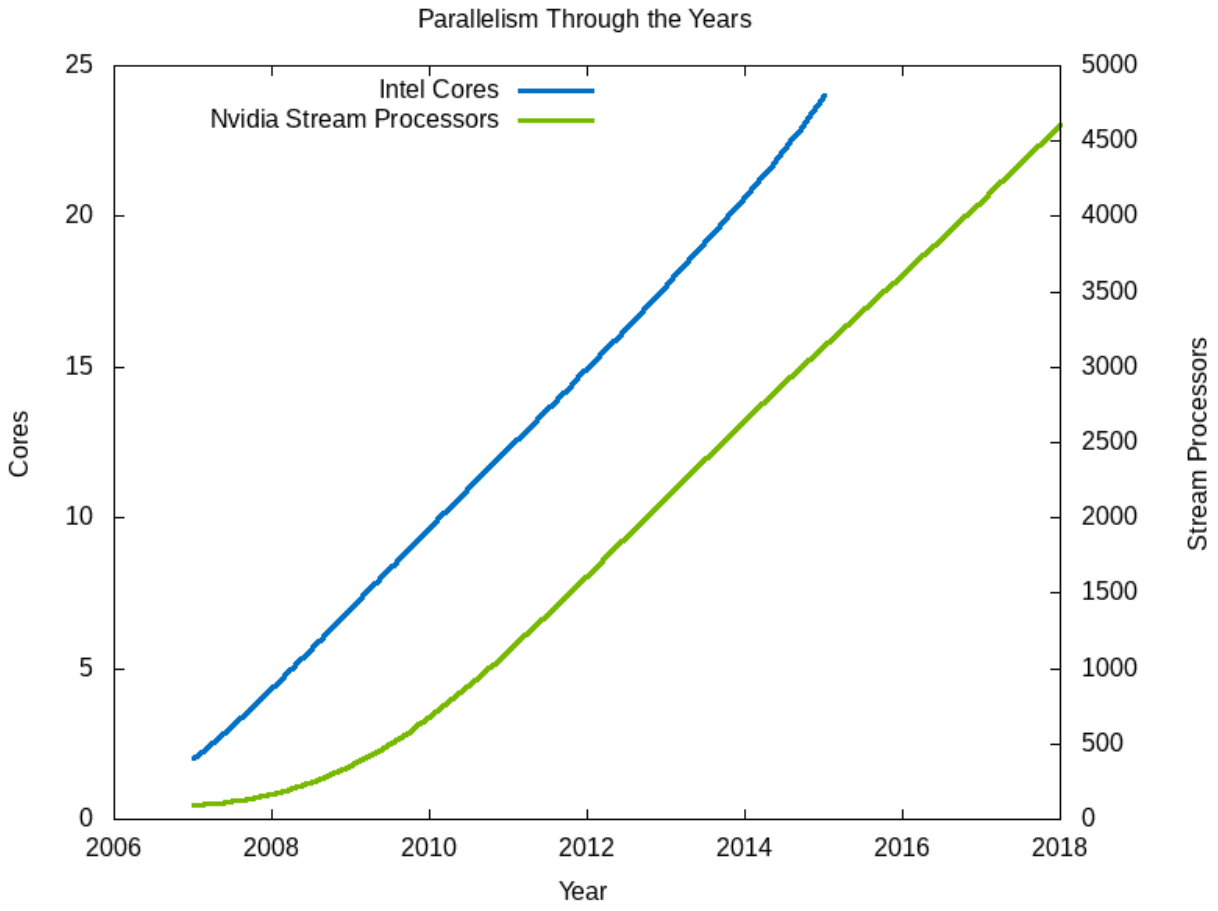



Figure 5.1: This graph shows a comparison in the growth of number of Intel CPU cores per chip and the number of Nvidia stream processors per GPU for each companies top of the line product. The trends are roughly even, with both increasing at similar rates, but Nvidia's GPU line has the greater overall number.

There have been attempts to speed up registration using a GPU. Olson (2009) presented a correlative scan matching approach and showed it could be offloaded to the GPU by using OpenGL shaders. Wolcott and Eustice (2014) also used OpenGL to localize a monocular camera stream into a LIDAR reflectivity prior map, by warping and calculating their mutual information cost function on the graphics card.

Other approaches to the data association stage of ICP have been attempted. Tamaki et al. (2010) used the GPU to perform a soft association between points. Newcombe et al. (2011) and Izadi et al. (2011) perform a projective association on the GPU, where the range image from time step $t$ is reprojected into the camera frame at timestep $t + 1$. This is the projective association approach explained in Chapter 1. The projective operator is independent to each point and therefore can be done simultaneously for all points. KD tree based associations are more complicated, Particularly

when building the data structure because the split process is dependent on the other points in the tree.

#### 5.2.1.1 GPU accelerated KD Tree

We have done initial work toward the goal of running registration algorithms on the GPU using KD tree based associations. Implemented with Nvidia's prorietery Compute Unified Device Architecture (CUDA) programming language, we developed a parallelized version of a $k$-dimensional (KD) tree build and query algorithms. Timing results are shown in Figure 5.2.

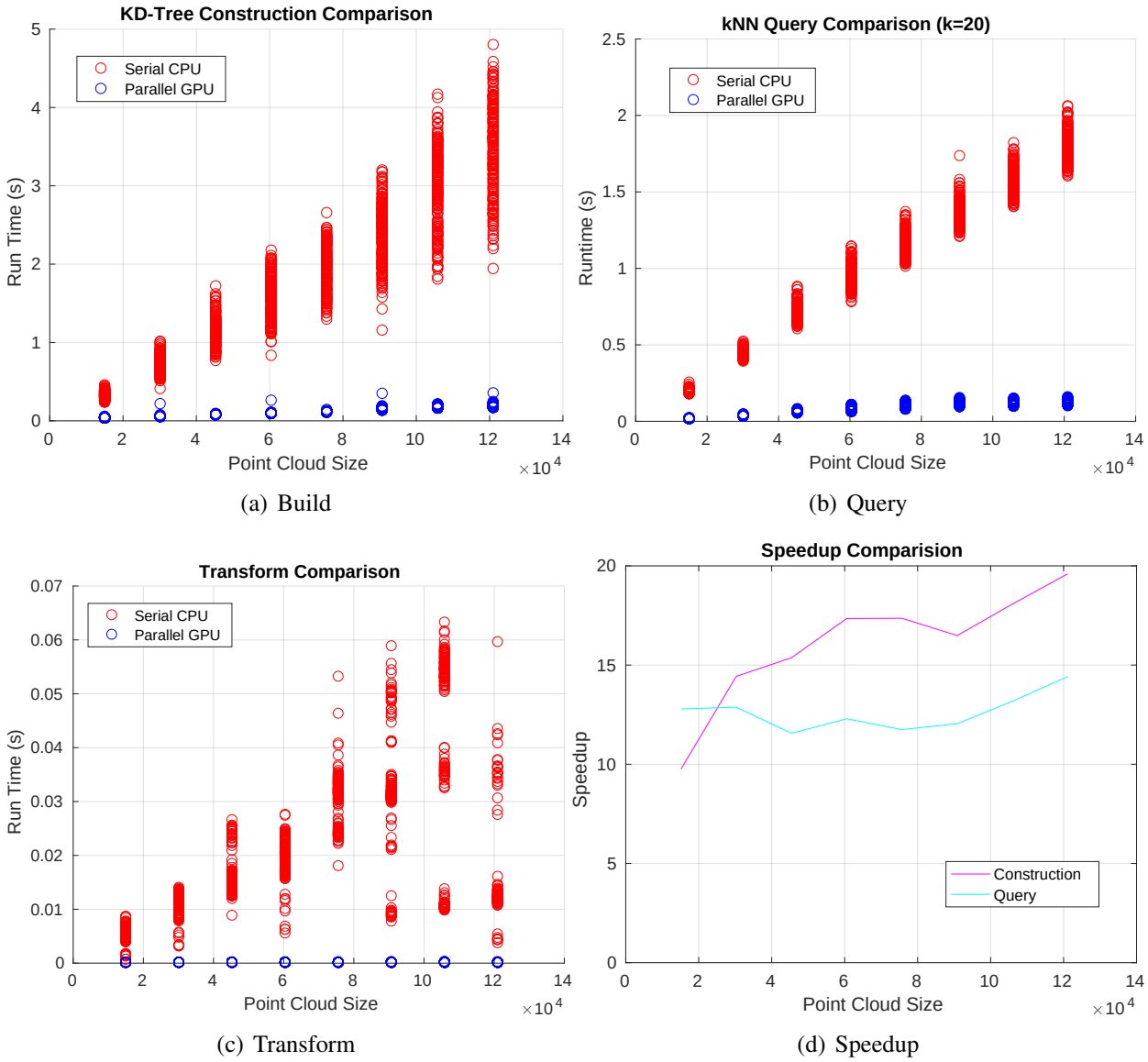
The recursive nature the KD tree construction task makes it difficult to parallelize. Many parallel algorithms first do the construction on the CPU and then transfer the tree to the GPU. The naive approach is to assign new threads to each new node that recursively gets created. The earliest nodes are the most computationally expensive to evaluate, and will the be the bottle neck in this naive method. We break the construction into two stages. In the first stage, when the nodes still have a large number of points that need to be partitioned, multiple threads are used to collaboratively partition the node. In the second stage, once the nodes have fewer points, each thread partitions the points in its own node. This second stage avoids expensive coordination when it starts dominating the run time of small nodes.

In our approach, each query point is assigned its own thread. Each thread traverses the tree in a branch-and-bound fashion. Each thread keeps a priority queue of maximum size $k$, and tree exploration is bounded by the maximum distance of a point in the queue.

When we applied this approach to the GICP algorithm we saw a six times speed up when compared to the same algorithm on the CPU, as shown in Figure 5.3. This shows the potential of porting registration algorithms to a GPU.

Finally, we used our GPU GICP algorithm in a real-time SLAM application using the NCLT dataset. For every scan that came in we ran GICP to determine the transformation from the previus frame to the current one producing a LIDAR odometry estimate. In addition, if the system had down time before the next frame came in, it performed registration on two frames that would maximize the information gain in factor graph slam estimate. We performed this for both the GPU and CPU implementations of the algorithms. The results are shown in Figure 5.4. It shows that GPU algorithm was able to add many more relative pose factors to the graphical SLAM estimate then the CPU algorithm was able to. This shows how improved runtime of a registration algorithm, even past the frame rate of the sensor, can aid down stream tasks.

Parallelizing the method described in Chapter 2 for semantic sensor registration would be a straight forward extension of the work we have presented above. This is because the residuals are similar to the original GICP formulation and would not have to be handled any differently then how we implemented it for our GPU GICP algorithm. The algorithm presented in Chapter 3 for $\mathcal{H}_k$-



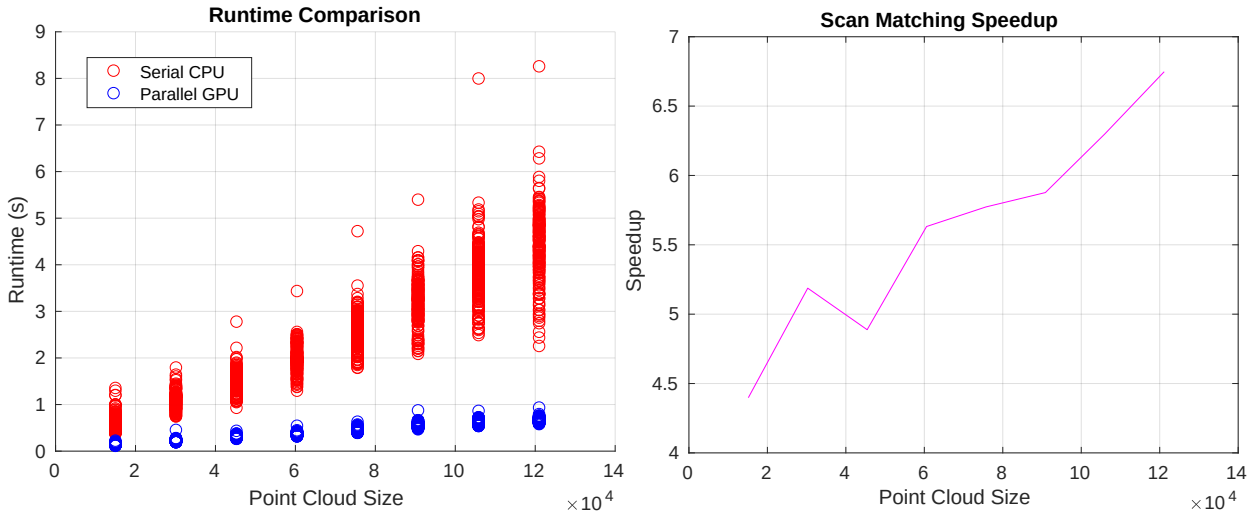

(a) Build

(b) Query

(c) Transform

(d) Speedup

Figure 5.2: A comparison of runtimes for the different aspects of scan matching. (a) and (b) show that the KD tree build and query stages weakly scale with the size of the point cloud. (c) is nearly constant on the GPU regardless of the point cloud size considering that there is no dependence between points for the transformation and therefore is trivially parallel.



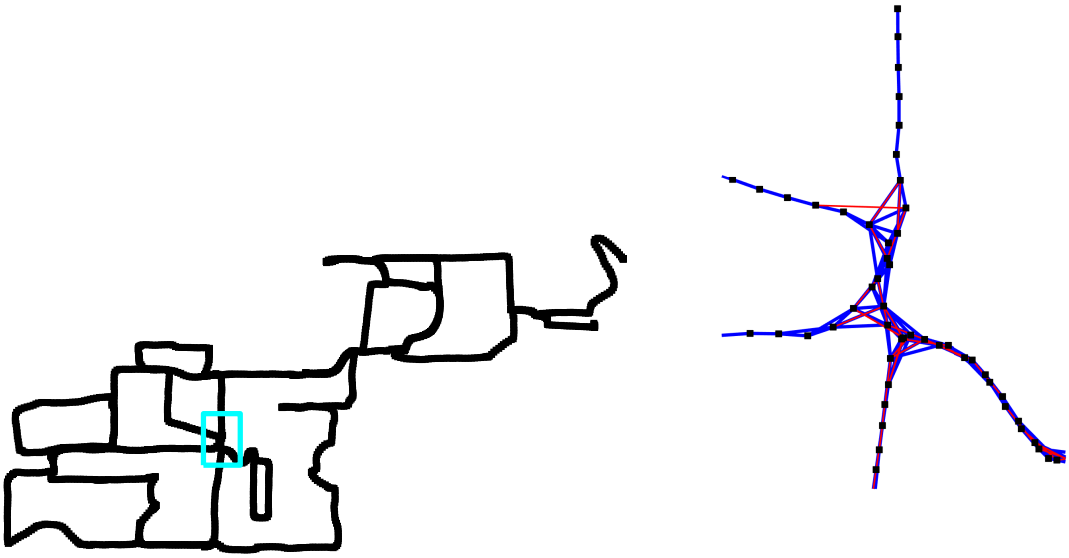

(a) Runtime (b) Speedup

Figure 5.3: Runtimes and Speedup of Segal, Haehnel, and Thrun (2009) implementation and the parallelized GPU version. Subfigure (a) Shows the run time distribution of the Parallel GPU and Serial CPU implementations of GICP. Subfigure (b) shows the average speed up of the Parallel implementation.

GICP-SE(3) would not be as straight forward. The training procedure presented is sequential and therefore has a lot dependencies between operations performed on each point. The batch method presented by Tipping (2001) would provide a better avenue for parallelization. It relies on constructing a large $n \times n$ kernel distance matrix though, which could exhaust the memory resources of a GPU. The evaluation and minimization of the algorithm similar to GICP, and therefore is less difficult to parallelize. Finally the focus of Chapter 4 was reformulating the registration problem as a MIP which we then solved using an off-the-shelf MIP solver. Many modern MIP solvers use threading to find solutions to the problem in a branch and bound (BnB) approach. It is an open question if SIMD parallelism is useful when solving these problems.

### 5.2.2 Curvature for Semantic Registration

Curvature of a surface in 3D space can loosely be defined by how much it varies from a flat plain. There are a variety of ways to quantify curvature, but in its many forms it has been found to be useful for computer graphics (Garland and Heckbert (1997); Griffin et al. (2012)), scene segmentation (Alshawabkeh, Haala, and Fritsch (2008)), and classification (Spek, Li, and Drummond (2017)). This is because the various parametrizations of curvature are local to that area of the surface that is being observed, and therefore are invariant to the viewpoint of the sensor and are invariant to transformations.



(a) Full graph (b) Parralel GPU vs Serial CPU

Figure 5.4: This figure illustrates the ability of our proposed system to process an increased number of scan matching factors in a online mapping setting. (a) shows the entire SLAM graph. (b) shows a zoomed-in view of the cyan rectangle. The black squares represent nodes in the SLAM graph. The blue lines between them indicate relative pose factors added by scan matching using our proposed system. The red lines indicate only those factors that would be added using a serial CPU scan matching system. The speed improvement our proposed system affords allows this application to process many more relative pose factors, ultimately leading to a more accurate SLAM solution.

Generalized Iterative Closest Point (GICP), the basis for Chapter 2, can be seen as fitting a plane to the local area of a surface by taking the numeric first derivative. Another approach could be to use curvature as found by the second derivative to align to point clouds.

#### 5.2.2.1 Background

Much of this background will follow the presentation in Elementary Differential Geometry by Andrew PressleyPressley (2010).

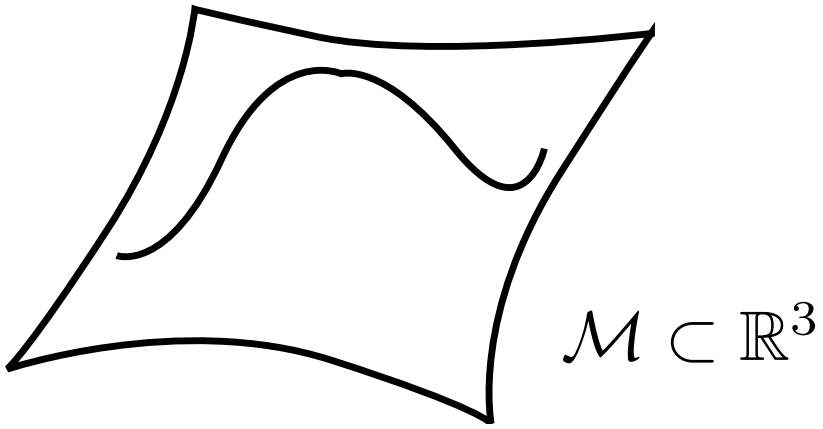



Figure 5.5: Illustration of surface in $\mathbb{R}^3$

Given a twice differentiable surface in 3D space, $\mathcal{M} \subset \mathbb{R}^3$, and a point on the that surface $\mathbf{x} \in \mathcal{M}$, by the definition of surface there is a parametrization at that point $\sigma(u, v) \in \mathcal{M}$ where $(u, v) \in \mathbf{R}^2$.

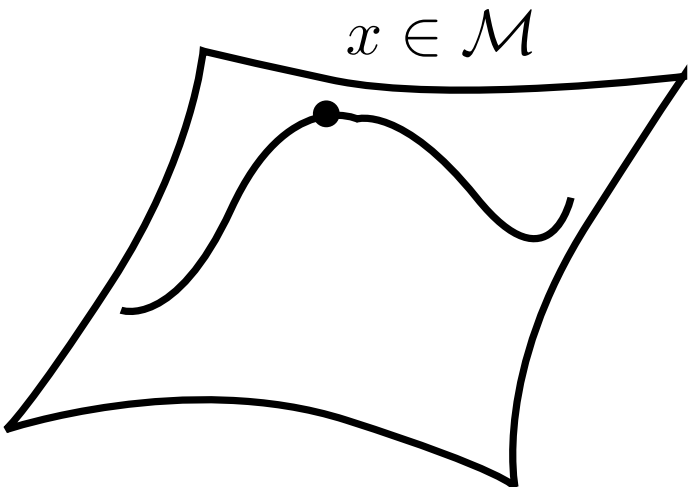



Figure 5.6: Illustration of a point on a surface

Let $\sigma_u = \frac{\partial \mathbf{u}}{\partial u}$ and $\sigma_v = \frac{\partial \mathbf{v}}{\partial v}$, the tangents space of the surface at point $\mathbf{x}$, or $T_{\mathbf{x}}\mathcal{M}$, is spanned by the vectors $\sigma_u$ and $\sigma_v$ and the normal is

$$\mathbf{N}_\sigma = \frac{\sigma_u \times \sigma_v}{||\sigma_u \times \sigma_v||} \tag{5.1}$$

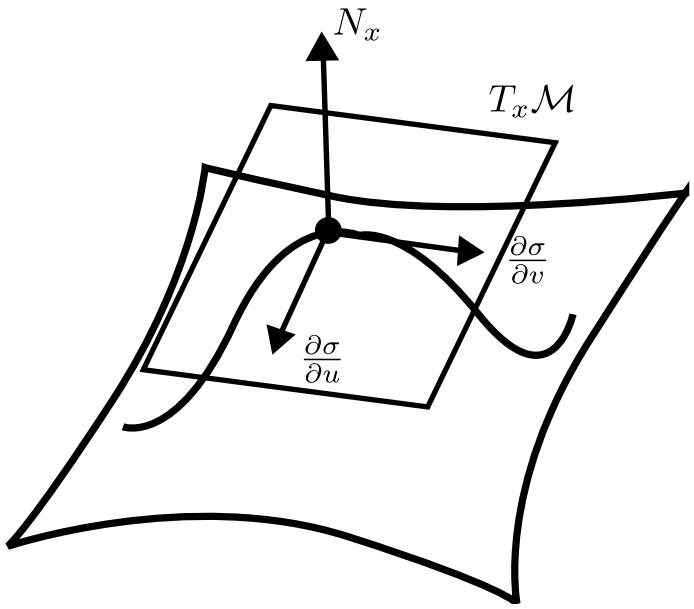



Figure 5.7: Illustration of a normal of surface

With a definition of the tangent space at the point, we can define the first fundamental form. The first fundamental form of $\mathcal{M}$ at $\mathbf{x}$ associates two tangent vectors $\mathbf{v}, \mathbf{w} \in T_{\mathbf{x}}\mathcal{M}$ with the scalar

$$\mathbf{I}(\mathbf{v}, \mathbf{w}) = \mathbf{v} \begin{bmatrix} E & F \\ F & G \end{bmatrix} \mathbf{w} \tag{5.2}$$

where $E = ||\sigma_u||^2$, $F = \sigma_u \cdot \sigma_v$, $G = ||\sigma_v||^2$. The first fundamental form allows for the measurement of lengths and areas on a surface, but is also used for some definitions of intrinsic curvatures. To see how a surface changes at point $\mathbf{x}$ with respect to the normal $\mathbf{N}$, the second fundamental form

$$\mathbf{II} = \begin{bmatrix} du & dv \end{bmatrix} \begin{bmatrix} L & M \\ M & N \end{bmatrix} \begin{bmatrix} du \\ dv \end{bmatrix} \tag{5.3}$$

is used, where $L = \sigma_{uu} \cdot \mathbf{N}$, $M = \sigma_{uv} \cdot \mathbf{N}$, $N = \sigma_{vv} \cdot \mathbf{N}$. By taking the Eigen decomposition of the second fundamental form $\mathbf{II}$ we can get the principal curvatures $\kappa_1, \kappa_2$ as the Eigenvalues, and the principal axis as the corresponding Eigenvectors. The principal curvature define the amount of curvature in the minimum and maximum axis, and are therefore viewpoint invariant. With them

we can define the Gaussian curvature

$$K = \kappa_1 \kappa_2 \tag{5.4}$$

or similarly using the first and second fundamental forms as

$$K = \frac{\det(\mathbf{II})}{\det(\mathbf{I})}. \tag{5.5}$$

The mean curvature is defined as

$$H = \frac{1}{2}(\kappa_1 + \kappa_2) \tag{5.6}$$

or with the first and second fundamental form as

$$H = \frac{1}{2}\operatorname{Tr}((\mathbf{II})(\mathbf{I}^{-1})). \tag{5.7}$$

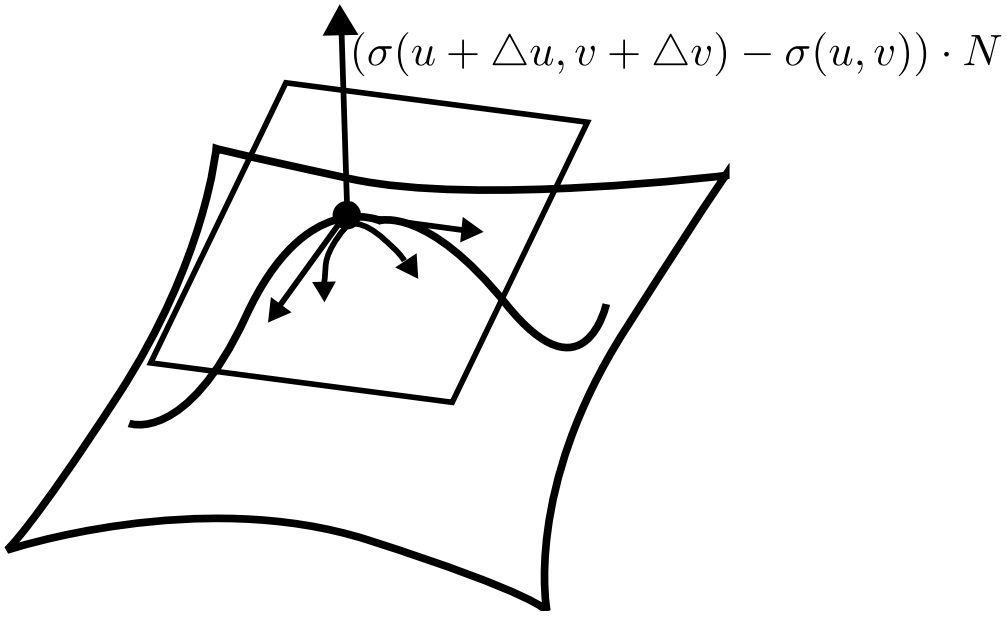



Figure 5.8: Illustration of a curve on a surface

These values have been shown to be useful or object recognitionSpek, Li, and Drummond (2017), and scene segmentationAlshawabkeh, Haala, and Fritsch (2008); Besl and Jain (1988), which shows its discriminative abilities. Our approach plans on using that ability to improve point cloud registration.

Table 5.1: Runtime of curvature estimation

| | Number of Point | KD-Tree Build (ms) | Curvature Estimation (ms) |
|---|---|---|---|
| KITTI odometry | 120000 | 84 | 15 |
| TUM RGB-D SLAM | 307200 | 127 | 34 |

#### 5.2.2.2 Potential Method

To be able to leverage the discriminative ability of the curvature we first need a way of computing it. To do this we note that in the parametrization of $\sigma(u, v)$ the second fundamental form consist of coefficients of the second order Taylor expansion at $\mathbf{x}$, rotated into the orientation of the tangent plane $T_{\mathbf{x}}\mathcal{M}$. In $\mathbb{R}^3$ this is equivalent to a quadric surface, which with homogeneous coordinates, $\widetilde{\mathbf{x}} = \begin{bmatrix} \mathbf{x} & 1 \end{bmatrix}^\top$, is defined as

$$\widetilde{\mathbf{x}}^\top Q \widetilde{\mathbf{x}} = 0 \tag{5.8}$$

where the matrix $Q$ is symmetric. Following the approach inSpek, Li, and Drummond (2017) we can fit a quadric $Q_i$ for point $\mathbf{x}_i$ minimizing the following error metric

$$\underset{Q_i}{\arg\min} \sum_{\mathbf{x}_j \in \mathtt{nn}(\mathbf{x}_i)} \left( \widetilde{\mathbf{x}}_j^\top Q_i \widetilde{\mathbf{x}}_j \right)^2 \tag{5.9}$$

and from $Q_i$ we can get $\kappa_1^i$ and $\kappa_2^i$. With those we propose to define some $\Sigma_\kappa$ to then use to determine the probability of association

$$p(\mathcal{I}|Q_i^s, Q_j^t) = \mathcal{N}\left( \begin{bmatrix} \kappa_1^i - \kappa_1^j \\ \kappa_2^i - \kappa_2^j \end{bmatrix}, \Sigma_\kappa \right) \tag{5.10}$$

which means it could then be used in the soft, EM, framework presented in Chapter 2.

#### 5.2.2.3 Initial Evaluation

An approach such as this would work well it the KITTI odometry and TUM RGB-D datasets. This would let us evaluate the performance on both light detection and ranging (LIDAR) and RGB-D data to see how the curvature estimates are affected by the different sensor modalities.

We have done some initial work to see if using curvature is feasible for use on real time registration tasks. Towards this goal, the curvature estimation algorithm has been implemented in CUDA on the graphics processing unit (GPU). We evaluated the curvature estimation approach on

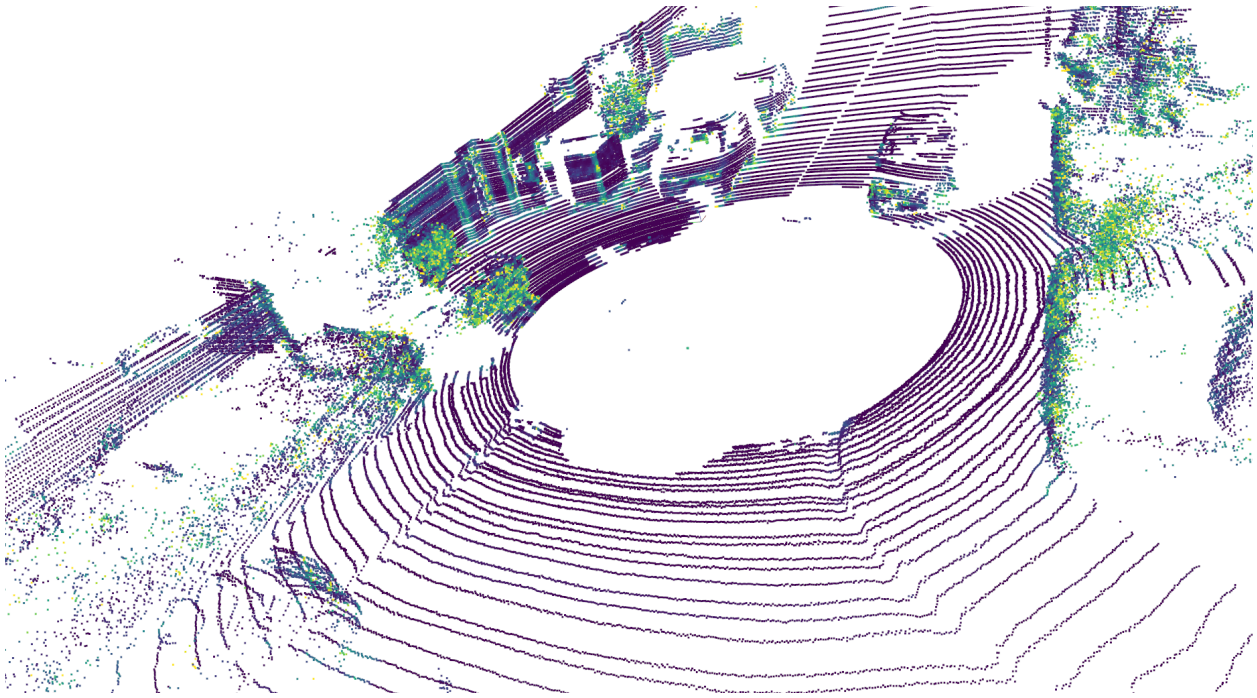


Figure 5.9: Principal curvature computed on LIDAR data from the KITTI odometry dataset. Colored by radius of the maximum curvature $\kappa_1$

data from the KITTI odometery and TUM RGB-D datasets. Run times are given in Table 5.1. An example scene from the KITTI odometry dataset is shown in Figure 5.9. With this initial evaluation, the run times combined with the qualitative results show that curvature estimation is fast and accurate enough to used in registration. Future work is needed to develop the curvature registration algorithm, and to figure out what approach to optimization is most appropriate.

Curvature, along with intensity and RGB information, is one of a few direct measurements available to a 3D sensor that is viewpoint invariant. Unlike intensity or RGB channels, though, curvature provides a stronger constraint on orientation when compared to another curvature measurement. Because of that, it has a lot of potential to improve the performance of registration algorithms. It does have drawbacks though. Determining the curvature requires some preprocessing time. And curvature is the derivative of what is being measured by the sensor, which is position. Because of that, noise in the sensor will be amplified when determining the curvature. Future work is needed to see if the benefits out way the drawbacks.

### 5.2.3 Dynamic Scenes

Most of the work presented in this document focuses on estimating the transformation of a sensor to data from some other frame of reference. In this way we are mainly addressing such

tasks as egomotion estimation and extrinsic calibration of multiple sensors. On top of that, we present systems that assume mostly static scenes, with any dynamic motion being handled by our robust estimation procedures. Few mobile robotic systems care solely about these kinds of transformations. They also need to know where other objects are in the environment and how they are moving so they can plan how to interact with them. In addition to being useful for planning, detecting and estimating dynamic objects in the environment can also improve the egomotion estimate by eliminating points that should not be included in the egomotion registration problem.

It is possible to estimate many rigid body transformations for all of the dynamic objects in the scene. This puts it in the middle of the egomotion to scene flow estimation spectrum. Scene flow estimation, such as that presented by Ushani et al. (2017), seeks to estimate the displacement of every voxel in the scene. Such estimation tasks can be intractable for large scenes. We could, instead, focus on estimating multiple rigid body transformations, one for each instance of a dynamic object.

Such an approach could leverage the method presented in Chapter 2, but instead use instance segmentation instead of semantic class segmentation. The inter-object data association could use the MIP approach presented in Chapter 4 where the smaller set of possible associations would make the optimization more efficient.

Adding objects to the registration problem also makes it a more complex filtering task. There would have to be a way to determine, frame to frame, when objects enter and leave the field of view of the sensor. If these challenges can be overcome, it would lead to a unified understanding of the robot state and the state of dynamic objects in the scene.

### 5.2.4 Eliminating Data Association Challenges

Many of the contributions presented in this thesis focus on overcoming failures in determining the correct data association. Data association problems are common for discretely sampled sensors such as LIDAR or red, gleen, blue, and depth (RGB-D) sensors. Solving the data associations is a challenge because, as was stated in Chapter 4, the space of possible associations grows exponentially with respect to the number of points. An alternative approach could be to try to eliminate the need to perform data association. Chapter 3 touches on some aspects of this idea, in that it regresses a function to avoid having to do an explicit data association. It only does so as a regularizer on top of the registration problem, and therefor just minimizes data association issue.

There has been some other work to eliminate data association from the registration problem. Bing Jian and Vemuri (2005) modeled the point cloud as a mixture of Gaussians, and formulated the registration problem as an alignment between two Gaussian mixtures, eliminate any explicit data association. Ghaffari et al. (2019) presented an approach that minimizes the distance between

two continuous functions that represent the source and the target data. What the approaches have in common is that the model measurements taken by the 3D sensor as a continuous function, which then no longer had discrete parts to associate to.

Potential future work could be to extend the approach presented Chapter 2 to be a continuous function. Potentially we could follow the approach of Bing Jian and Vemuri (2005) and model the point cloud as mixture of Gaussians and add to the distribution the semantic probabilities estimated by the classifier. That way, when we align the distributions, we will also be making the classification results consistent.

### 5.2.5 Bespoke Optimization Approaches

The work presented in this thesis presents new formulations to registration and finds the solution using off the shelf solvers such as Ceres-Solver and Gurobi. To get the most efficiency and accuracy, switching to a bespoke optimization strategy might hold the most benefit. Briales and Gonzalez-Jimenez (2017) present an approach to registration that uses convex relaxation, for which they develop a custom solver. Yang, Li, and Jia (2013) presented a custom BnB solver to find the global solution to the point cloud registration problem. Parra Bustos, Chin, and Suter (2014) presented a custom BnB solver for stereographic projections. Similarly for multi-resolution search, Olson (2009) presented a bespoke approach to finding the minimum of his correlative scan matching formulation, and Wolcott and Eustice (2017b) similarly presented a solver to optimizer his Gaussian mixture maps. All of these methods present bespoke optimizers to find the optimal solution to the proposed registration cost functions so that they can be solved in an efficient manner.

There are promising reasons why a bespoke optimization technique would be useful for 2D to 3D registration like what is proposed in Chapter 4. For a custom BnB solver for a 2D to 3D registration, we would want to bound where a potential point would end up in an image under a set of transformations, $\{\mathbf{T}\}$, this is illustrated in Figure 5.10

#### 5.2.5.1 Bounding Image Region

To bound which pixels $u, v$ the point can correspond to for an interval of transformations. Yang, Li, and Jia (2013) found a bound in translation $\gamma_t$ and rotation $\gamma_r$ such that

$$\forall \mathbf{T} \in \{\mathbf{T}\}, ||\mathbf{T}\{x\} - x_0|| < \gamma_t + \gamma_r.$$

This means the possible positions of $x$ transformed by and the interval $\{\mathbf{T}\}$ is bounded by a sphere in $\mathbb{R}^3$ of radius $\gamma_r + \gamma_t$.

In contrast, we could bound the pixel values by finding the min and max of points where the

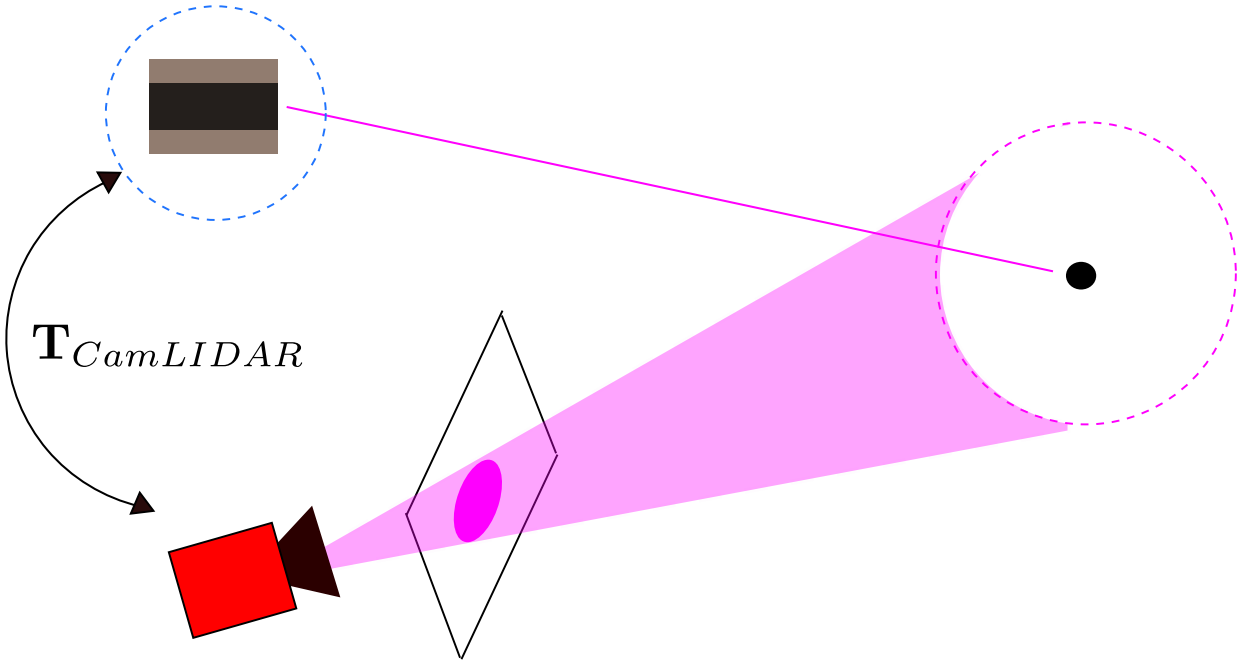



Figure 5.10: An illustration of how an interval in the transformation $\{\mathbf{T}\}$ can affect where a point detected by the LIDAR sensor ends up in the camera Image.

projection of the point onto the image is zero, or $I'(\cdot)$, or the edges of the transformation set $\{\mathbf{T}\}$. These would be the critical points of the projection of the point onto the image.

#### 5.2.5.2 2D Range Maximum Query

Now that we have a possible range of pixel values that a point $x_i$ may correspond to, we need an efficient way to find the max in that interval to bound the cost function. Each image will receive many range queries while search over the transformation interval. This falls under a range of problems known as Range Minimum Queries that are well studied in the 1D case, and which there is some work in the 2D case.

We could use a data structure called a sparse table, for which we compute every possible query where the height and width is a power of 2. This allows us to only need to find the minimum of 4 elements of the sparse table for any potential query. This approach therefore has $\mathcal{O}(c)$, or constant, query time and $\mathcal{O}(NM \log N \log M)$ preprocessing time and memory for and $N$ by $M$ array. An illustration of how a sparse table works is shown in Figure 5.12.

Sparse tables are a generalization of the strategy used by multi-resolution search. In multi resolution search, such as what is proposed by Olson (2009) and by Wolcott and Eustice (2017b), use a smaller subset of cononical arrays. They couple this to a fixed sized interval of translation, and sample over a discrete set of rotations. They do this because after applying rotation, translation

is axis aligned with the target reference frame. This is shared with our 2D to 3D formulation, in that after applying the estimated rotation, the translation of the point will be axis aligned with the image. Unfortunately, because of the effect of the depth while projecting a 3D point onto the image plane, a fixed sized translation does not lead to a fixed size bounding box in the image for all possible point locations. Despite that though, we could combine the discrete rotation sampling used in multi resolution searches with a sparse table data structure to quickly search over the possible translations.

#### 5.2.5.3 Other Approaches

Beyond custom BnB there are other approaches to optimization that may be useful. Briales and Gonzalez-Jimenez (2017) presented work that found the global optimal solution to the point to line and point to plane registration cost function using convex relaxation. There are other ways of exploiting the structure of the $\mathrm{SE}(3)$ group. Izatt, Dai, and Tedrake (2017) developed a way of representing the nonconvex constraint within the $\mathrm{SO}(3)$ group as a piecewise linear constraint for use in a MIP formulation of the problem. They also showed it could work in problems outside of registration, such as inverse kinematics problems (Dai, Izatt, and Tedrake).

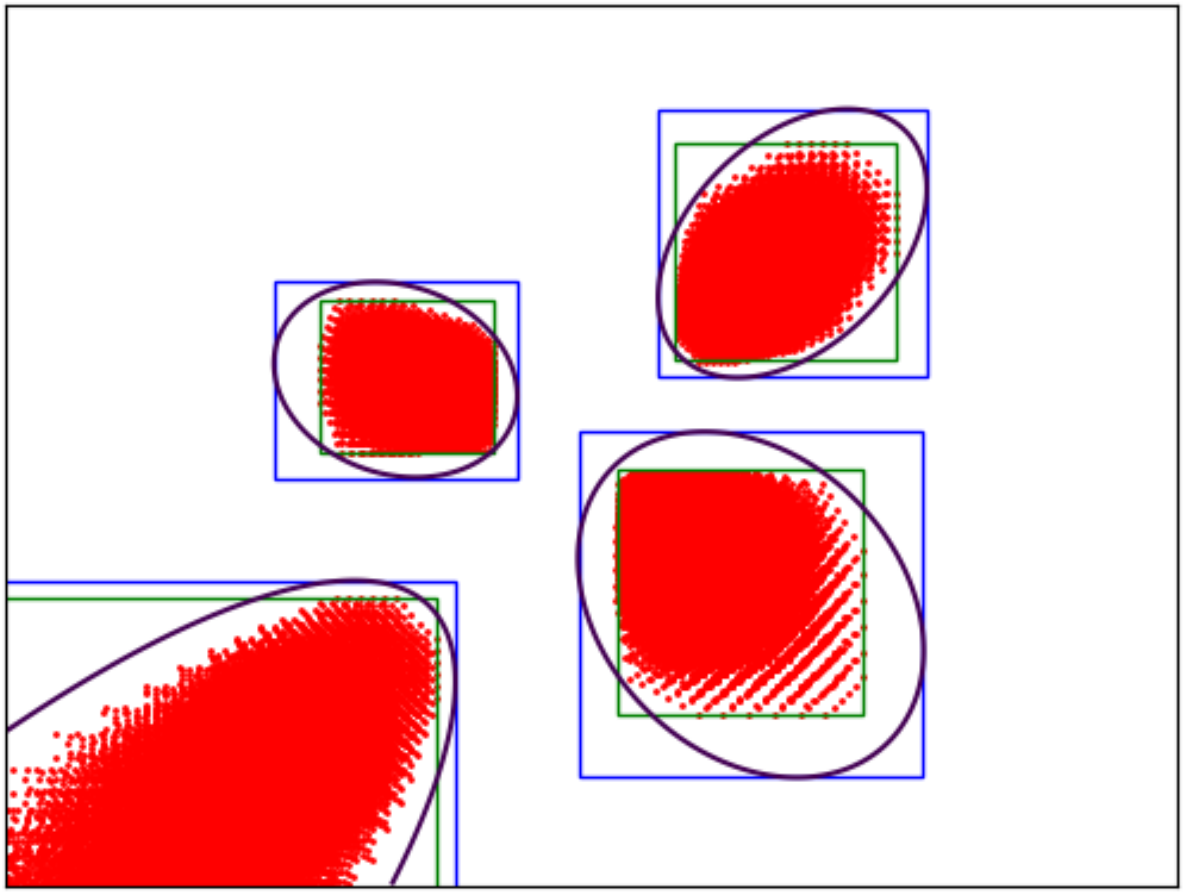


Figure 5.11: In this figure we sampled transformations in an interval to generate the possible affects of that interval on 4 points (shown in red). We then show the encapsulating sphere of radius $\gamma$ as predicted by the method proposed by Yang, Li, and Jia (2013) projected onto the image (shown in black). To work in an efficient manner with the sparse table data structure, the query needs to be an axis aligned rectangle. The rectangular region defined by the min and max $u$ and $v$ of that sphere is in blue. Finally we show the rectangular bound derived from the critical points in green. This shows the critical point method is more strict then the bounding sphere.

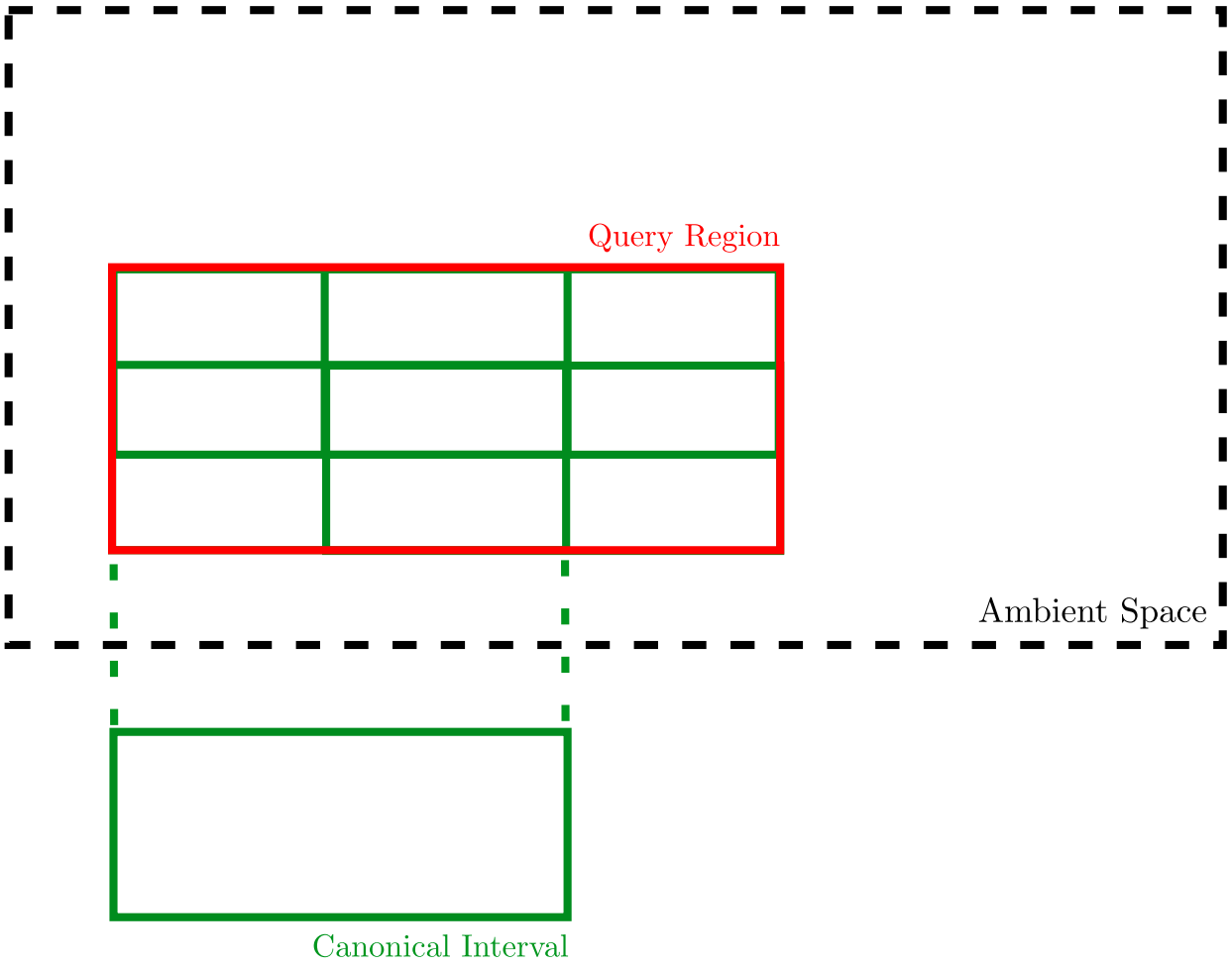



Figure 5.12: This illustrates how a sparse table is able to quickly compute the maximum of any subset of a 2D array. By storing the max of all cononical arrays (any array whose sides are a power of 2) and subset of the array can reproduced by finding the max of four canonical arrays.

# APPENDICES

# APPENDIX A

# Lie Group Notations

In this appendix, we explain the notation used throughout this thesis as well as the required preliminaries. Matrices are capitalized in bold, such as in $\mathbf{X}$, and vectors are in lower case bold type, such as in $\mathbf{x}$. Vectors are column-wise and $1:n$ means the integers from 1 to $n$. $\text{vec}(x_1, \ldots, x_n)$ denotes a vector such as $\mathbf{x}$ constructed by stacking $x_i$, $\forall\ i \in \{1:n\}$. An alphabet such as $\mathcal{X}$ denotes a set. The Euclidean norm is shown by $\|\cdot\|$. $\|\mathbf{e}\|^2_{\mathbf{\Sigma}} \triangleq \mathbf{e}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{e}$. The $n$-by-$n$ identity matrix is denoted by $\mathbf{I}_n$. $\mathbf{0}_n$ denotes the vector of zeros with dimensions $n$.

Thorough details of the covered topics in this section are available in Absil, Mahony, and Sepulchre (2009); Chirikjian (2011); Murray et al. (1994). The general linear group of degree $n$, denoted by $\text{GL}_n(\mathbb{R})$, is the set of all $n \times n$ nonsingular real matrices, where the group binary operation is the ordinary matrix multiplication. The 3D special orthogonal group, denoted by

$$\text{SO}(3) = \{\mathbf{R} \in \text{GL}_3(\mathbb{R}) | \mathbf{R}\mathbf{R}^\mathsf{T} = \mathbf{I}_3, \det \mathbf{R} = +1\},$$

is the rotation group on $\mathbb{R}^3$. The 3D special Euclidean group, denoted by

$$\text{SE}(3) = \{\mathbf{T} = \begin{bmatrix} \mathbf{R} & \mathbf{p} \\ \mathbf{0}_3^\mathsf{T} & 1 \end{bmatrix} \in \text{GL}_4(\mathbb{R}) | \mathbf{R} \in \text{SO}(3), \mathbf{p} \in \mathbb{R}^3\},$$

is the group of rigid transformations on $\mathbb{R}^3$. The Lie algebra (tangent space at the identity together with Lie bracket) of $\text{SO}(3)$, denoted by $\mathfrak{so}(3)$, is the set of $3 \times 3$ skew-symmetric matrices such that for any $\boldsymbol{\omega} \triangleq \text{vec}(\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$: $\boldsymbol{\omega}^\wedge \triangleq \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix}$ and $(\boldsymbol{\omega}^\wedge)^\vee = \boldsymbol{\omega}$. The Lie algebra of $\text{SE}(3)$, denoted by $\mathfrak{se}(3)$, can be identified by $4 \times 4$ matrices such that for any $\boldsymbol{\omega}, \mathbf{v} \in \mathbb{R}^3$ and $\boldsymbol{\xi} \triangleq \text{vec}(\boldsymbol{\omega}, \mathbf{v}) \in \mathbb{R}^6$: $\boldsymbol{\xi}^\wedge \triangleq \begin{bmatrix} \boldsymbol{\omega}^\wedge & \mathbf{v} \\ \mathbf{0}_3^\mathsf{T} & 0 \end{bmatrix}$. The exponential map $\exp : \mathfrak{se}(3) \to \text{SE}(3)$ can be used to map a member of $\mathfrak{se}(3)$ around a neighborhood of zero to a member of $\text{SE}(3)$ around a neighborhood of the identity. The logarithm map is the inverse, i.e. $\log : \text{SE}(3) \to \mathfrak{se}(3)$, and $\exp(\log(\mathbf{T})) = \mathbf{T}$. Now we can define the difference between a transformation $\mathbf{T} \in \text{SE}(3)$ and its

estimate with a small perturbation $\hat{\mathbf{T}} \in \mathrm{SE}(3)$ as Barfoot and Furgale (2014); Chirikjian (2011):

$$\boldsymbol{\epsilon}^{\wedge} = \log(\hat{\mathbf{T}}\mathbf{T}^{-1})$$

where $\boldsymbol{\epsilon}^{\wedge} \in \mathfrak{se}(3)$. To define the norm of the error term, we exploit the fact that $\mathfrak{se}(3)$ is isomorphic to $\mathbb{R}^6$, i.e. $\boldsymbol{\epsilon}^{\wedge} \mapsto \boldsymbol{\epsilon} \in \mathbb{R}^6$ using the $\vee$ operator. Thus $\|\boldsymbol{\epsilon}\| = \|\log(\hat{\mathbf{T}}\mathbf{T}^{-1})^{\vee}\|$, and we define $\|\boldsymbol{\epsilon}\|^2_{\boldsymbol{\Sigma}} \triangleq \boldsymbol{\epsilon}^{\top}\boldsymbol{\Sigma}^{-1}\boldsymbol{\epsilon}$.

# APPENDIX B

# Manifold Optimization

This appendix is meant to provide intuition into the optimization techniques used in Chapter 2 and Chapter 3 and follows the presentation in Optimization Algorithms on Matrix Manifolds by Absil, Mahony, and Sepulchre (2009) and people interested in a more detailed explanation should look there.

There are many problems where we are interested in optimizing a function $f(\cdot)$ that maps a point on a manifold, $m \in \mathcal{M}$, to a real value, or $f : \mathcal{M} \mapsto \mathbb{R}$. Most gradient based approaches to optimization were designed to work with functions that map from a Euclidean space $\mathbb{R}^n$ to a real value. Fortunately manifolds have a structure that locally makes then similar to Euclidean spaces. Each point $m$ in a Manifolds has a neighborhood, $\mathcal{U}$, that is a subset of the manifold which has a one to one mapping to the Eucilidan space $\mathbb{R}^d$. This mapping is called a coordinate chart, $\rho(\cdot)$, $\rho : \mathcal{U} \mapsto \mathbb{R}^d$. Therefore, in the subset $\mathcal{U}$ of manifold $\mathcal{M}$ we have a real valued function $f \circ \rho^{-1}$ that maps from a Euclidean space to real values, $f \circ \rho^{-1} : \mathbb{R}^d \mapsto \mathbb{R}$.

A lot of structures that come up in the study of robotic systems are Manifolds. The most basic is Euclidean spaces themselves. While it is trivial, every member of a Euclidean space has a neighborhood that is also Euclidean. The special orthogonal group, $\mathrm{SO}(n)$, that represents rotations is also a manifold, and the special euclidean group, $\mathrm{SE}(n)$, that represents transformations is a Manifold that is the product of the special orthogonal group and the Euclidean space $\mathbb{R}^n$ that represents translation. The n-sphere $S^n$ represents a circle in two dimensions and a sphere in three. All of these structures are manifolds with the structure described above.

To solve optimization problems on a manifold, we follow the approach presented in Optimization Algorithms on Matrix Manifolds. Their approach can be described by the phrase "lift, solve, retract".

## B.1 Lift

Lifting describes the process of taking the real valued objective function $f$ with range on the manifold $\mathcal{M}$ to be have range in a Euclidean space. This is done using the composite function $f \circ \rho^{-}1$.

## B.2 Solve

Solve step is the same that is used in Euclidean spaces. It is the process of finding some solution $\hat{\xi}$ that minimizes the function $f \circ \rho^{-}1$. This can be done in a variety of ways. Both line search and trust region methods are proposed in Optimization Algorithms over Matrix Manifolds.

## B.3 Retract

Once we have a solution in the Euclidean space attached to the neighborhood of a point on the manifold, we need to bring it back down onto the manifold. This is called a retraction, and gives us a solution that is a member of the manifold, $\hat{m} = \rho^{-1}\left(\hat{\xi}\right)$.

## B.4 Instantiation for $\mathrm{SE}(3)$

For $\mathrm{SE}(3)$, the exponential map can serve as this retraction, and we solve the optimization problem by iteratively lifting (logarithm map) the cost function to the tangent space, solving the reparameterized problem, and then mapping the updated solution back to the original space using the retraction. For this work, we use the open source library Ceres Solver by Agarwal, Mierle, and Others. Using its local parametrizations, we can solve the nonlinear least squares problem by going to and from the tangent space of $\mathrm{SE}(3)$. Manopt (Boumal et al. (2014)) provides another implementation of this procedure.

# APPENDIX C

# Mixed-Integer Programming

A mixed-integer program is a optimization program where some of the variables are integer valued. Particularly for this thesis, we were interested in mixed-integer linear programs which have the form

$$\begin{aligned} \text{maximize} & \mathbf{c}^\top \mathbf{x} + \mathbf{d}^\top \mathbf{y} \\ \text{s.t.} & \mathbf{A}\mathbf{x} + \mathbf{E}\mathbf{y} \leq \mathbf{b} \\ & \mathbf{x}_{\min} \leq \mathbf{x} \leq \mathbf{x}_{\max} \\ & \mathbf{y} \in \mathbb{Z} \end{aligned} \tag{C.1}$$

which means the vector $\mathbf{y}$ is constrained to be integer valued. Solving problems for integer valued variables is NP-complete, in that the solution can only be found through some variation of a brute-force search.

## C.1 Solving Mixed-Integer Programs

Developing methods to solve MIP was not the focus of this thesis. For the work presented in Chapter 4, an off the shelf solver (Gurobi) was used. But to understand MIP, it is important to know some of the algorithms used to solve them.

The BnB approach is a standard search technique that can be applied to a MIP. This approach subdivides the feasible region and bounds it by replacing the integer constraint with a real value constrain forming a linear program. Any linear program formed by relaxing a integer constraint will have a solution that is greater then or equal to the equivalent MIP solution.

Cutting-plane similarly works by relaxing the integer constraint. Once a solution to the associated linear program is found. If the solution is not integer valued, then an inequality constraint is added with the linear program solution on one side, and all feasible integer solutions on the other, effectively removing the linear program solution.

A combination of the two approaches above has been used, called branch and cut. In this framework, feasible regions are subdivided like they are in branch and bound. When the subregion

is relaxed to a linear program to determine the bound, it is also analyzed to see if an effective cutting plane can be applied.

Finally, heuristics can be used to find feasible solutions. The solutions can then be used as lower bound on a branch and bound or branch and cut strategy, limiting the number of regions that need to be evaluated.

# APPENDIX D

# Software Repositories

In this appendix we will go over some of the software repositories that we have made available which contain implementations of the contributions presented in this thesis.

## D.1 Semantic Iterative Closest Point

**https://bitbucket.org/saparkison/perl_registration**

This repository includes a c++ implementation of the method proposed in Chapter 2. It is build using cmake with the following command

```
$ mkdir build
$ cd build
$ cmake ../
$ make
```

We also include code to evaluate the results on the KITTI odometry dataset, and a visualization tool to view the results. In addition, this repository includes an implementation of the KD tree data structure on the GPU, a GPU GICP algorithm, and a version of the proposed method that runs on the GPU. It has various dependencies such as PCL, Ceres Solver, and the cuda developement libraries.

## D.2 Intensity Regularized Registration

**https://bitbucket.org/saparkison/rkhs_gicp**

This repository includes an c++ implementation of the method proposed in Chapter 3 and code to evaluate the proposed method on the KITTI odometry dataset and TUM RGB-D dataset. It is built using bazel with the command

```
$ bazel build ...
```

This includes our c++ version of the sequential training method proposed by Tipping, Faul et al. (2003) (originally released as Matlab code) and our implementation of the multi-channel GICP algorithm. Its dependencies are managed by bazel and do not have to installed before building.

## D.3 MIP Registration formulation

**https://bitbucket.org/saparkison/plucker_line_mip**

This repository includes our Julia implementation of the method proposed in Chapter 4. Or approach uses the JuMP library that provides an abstraction to interface off the Gurobi Optimization library, but which can be used with several other open source MIP solvers. It also includes our implementation of the singular value decomposition (SVD) approach proposed by Přibyl, Zemčík, and Čadík (2016) (originally released as Matlab code) and evaluation code for the VGG dataset. This repository additionally includes our proposed approach to extracting linear semantic features from images. It depends on several other Julia packages, but those are handled by the Julia package manager.

# BIBLIOGRAPHY

# BIBLIOGRAPHY


P.-A. Absil, R. Mahony, and R. Sepulchre. Optimization algorithms on matrix manifolds. Princeton University Press, 2009.

S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org.

H. Alhaija, S. Mustikovela, L. Mescheder, A. Geiger, and C. Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. International Journal of Computer Vision (IJCV), 2018.

Y. Alshawabkeh, N. Haala, and D. Fritsch. Range image segmentation using the numerical description of the mean curvature values. In The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. ISPRS Congress, page 533, 2008.

S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In Proceedings of the IEEE International Conference on Computer Vision, pages 2425–2433, 2015.

S. Y. Bao and S. Savarese. Semantic structure from motion. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., pages 2025–2032. IEEE, 2011.

T. D. Barfoot and P. T. Furgale. Associating uncertainty with three-dimensional poses for use in estimation problems. IEEE Trans. Robot., 30(3):679–693, 2014.

C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics, 28(3):24, 2009.

A. Bartoli and P. Sturm. The 3d line motion matrix and alignment of line reconstructions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 1, pages I–I. IEEE, 2001.

A. Bartoli and P. Sturm. Structure-from-motion using lines: Representation, triangulation, and bundle adjustment. Computer vision and image understanding, 100(3):416–441, 2005.

A. Berlinet and C. Thomas-Agnan. Reproducing kernel Hilbert spaces in probability and statistics. Kluwer Academic, 2004.

P. J. Besl and R. C. Jain. Segmentation through variable-order surface fitting. IEEE Trans. Pattern Anal. Mach. Intell., 10(2):167–192, Mar. 1988. ISSN 0162-8828. doi: 10.1109/34.3881. URL http://dx.doi.org/10.1109/34.3881.

P. J. Besl and N. D. McKay. Method for registration of 3-d shapes. In Sensor Fusion IV: Control Paradigms and Data Structures, volume 1611, pages 586–607. International Society for Optics and Photonics, 1992.

K. K. S. Bhat and J. Heikkilä. Line matching and pose estimation for unconstrained model-to-image alignment. In 2014 2nd International Conference on 3D Vision, volume 1, pages 155–162, Dec 2014. doi: 10.1109/3DV.2014.27.

P. Biber and W. Straßer. The normal distributions transform: A new approach to laser scan matching. In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., volume 3, pages 2743–2748. IEEE, 2003.

Bing Jian and B. C. Vemuri. A robust algorithm for point set registration using mixture of gaussians. In Proceedings of the IEEE International Conference on Computer Vision, volume 2, pages 1246–1251 Vol. 2, Oct 2005. doi: 10.1109/ICCV.2005.17.

C. M. Bishop. Pattern recognition and machine learning. springer, 2006.

N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre. Manopt, a Matlab toolbox for optimization on manifolds. Journal of Machine Learning Research, 15:1455–1459, 2014. URL http://www.manopt.org.

S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic SLAM. In Proc. IEEE Int. Conf. Robot. Automat., pages 1722–1729, 2017.

J. Briales and J. Gonzalez-Jimenez. Convex global 3D registration with Lagrangian duality. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., pages 4960–4969, 2017.

G. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. European Conf. on Computer Vision, pages 44–57, 2008.

D. Campbell, L. Petersson, L. Kneip, and H. Li. Globally-optimal inlier set maximisation for simultaneous camera pose and feature correspondence. In Proceedings of the IEEE International Conference on Computer Vision, volume 1, 2017.

N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice. University of Michigan North Campus long-term vision and lidar dataset. International Journal of Robotics Research, 35(9):1023–1035, 2015.

R. O. Castle, D. J. Gawley, G. Klein, and D. W. Murray. Towards simultaneous recognition, localization and mapping for hand-held and wearable cameras. In Proc. IEEE Int. Conf. Robot. Automat., pages 4102–4107, 2007.

A. Censi. An ICP variant using a point-to-line metric. In Proc. IEEE Int. Conf. Robot. Automat., pages 19–25. IEEE, 2008.

A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012, 2015.

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, Atrous convolution, and fully connected CRFs. 2016.

Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In Proc. IEEE Int. Conf. Robot. Automat., pages 2724–2729. IEEE, 1991a.

Y. Chen and G. Medioni. Object modeling by registration of multiple range images. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 2724–2729. IEEE, 1991b.

G. S. Chirikjian. Stochastic Models, Information Theory, and Lie Groups, Volume 2: Analytic Methods and Modern Applications. Springer Science & Business Media, 2011.

S. Choudhary, A. J. Trevor, H. I. Christensen, and F. Dellaert. SLAM with object discovery, modeling and mapping. In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pages 1018–1025, 2014.

J. Civera, D. Gálvez-López, L. Riazuelo, J. D. Tardós, and J. Montiel. Towards semantic SLAM using a monocular camera. In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pages 1277–1284, 2011.

M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., 2016.

C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. arXiv preprint arXiv:1301.3572, 2013.

H. Dai, G. Izatt, and R. Tedrake. Global inverse kinematics via mixed-integer convex optimization. International Journal of Robotics Research, page 0278364919846512.

P. David, D. DeMenthon, R. Duraiswami, and H. Samet. Simultaneous pose and correspondence determination using line features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, volume 2, pages II–II. IEEE, 2003.

P. David, D. Dementhon, R. Duraiswami, and H. Samet. Softposit: Simultaneous pose and correspondence determination. International Journal of Computer Vision, 59(3):259–284, 2004.

I. Dunning, J. Huchette, and M. Lubin. Jump: A modeling language for mathematical optimization. SIAM Review, 59(2):295–320, 2017. doi: 10.1137/15M1020575.

G. Elbaz, T. Avraham, and A. Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on, pages 2472–2481. IEEE, 2017.

J. Engel, J. Stueckler, and D. Cremers. Large-scale direct slam with stereo cameras. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2015.

D. Ferstl, G. Riegler, M. Ruether, and H. Bischof. Cp-census: A novel model for dense variational scene flow from rgb-d data. In Proceedings of the British Machine Vision Conference, Nottingham, UK, September 2014.

M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Commun. ACM, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL http://doi.acm.org/10.1145/358669.358692.

S. Fontana, T. Hinzmann, and G. Agamennoni. Iterative Probabilistic Data Association. https://github.com/ethz-asl/robust_point_cloud_registration, 2016. [Online; accessed 30-January-2018].

R. Y. S. Gabriel Agamennoni, Simone Fontana and D. G. Sorrenti. Point clouds registration with probabilistic data association. In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pages 4092–4098. IEEE, 2016.

M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In Proceedings of the 24th annual conference on Computer graphics and interactive techniques, pages 209–216. ACM Press/Addison-Wesley Publishing Co., 1997.

A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In Asian Conf. Computer Vision, 2010.

A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., 2012.

A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. International Journal of Robotics Research, 2013.

M. Ghaffari, W. Clark, A. Bloch, R. M. Eustice, and J. W. Grizzle. Continuous direct sparse visual odometry from RGB-D images. In Proceedings of the Robotics: Science & Systems Conference, Freiburg, Germany, June 2019.

R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 580–587, 2014.

R. Gomez-Ojeda, F. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez. Pl-slam: A stereo slam system through the combination of points and line segments. 35(3):734–746, June 2019.

S. Granger, X. Pennec, and A. Roche. Rigid point-surface registration using an em variant of ICP for computer guided oral implantology. In W. J. Niessen and M. A. Viergever, editors, Medical Image Computing and Computer-Assisted Intervention – MICCAI 2001, pages 752–761, Berlin, Heidelberg, 2001. Springer Berlin Heidelberg.

A. Gressin, C. Mallet, J. Demantké, and N. David. Towards 3D lidar point cloud registration improvement using optimal neighborhood knowledge. ISPRS journal of photogrammetry and remote sensing, 79:240–251, 2013.

W. Griffin, Y. Wang, D. Berrios, and M. Olano. Real-time gpu surface curvature estimation on deforming meshes and volumetric data sets. IEEE Transactions on Visualization and Computer Graphics, 18(10):1603–1613, Oct 2012. ISSN 1077-2626. doi: 10.1109/TVCG.2012.113.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., pages 770–778, 2016.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., pages 2980–2988. IEEE, 2017.

Y. He, J. Zhao, Y. Guo, W. He, and K. Yuan. Pl-vio: Tightly-coupled monocular visual–inertial odometry using point and line features. Sensors, 18(4):1159, 2018.

B. K. Horn and B. G. Schunck. Determining optical flow. Artificial Intelligence, 17(1):185–203, 1981.

M. Hornacek, A. Fitzgibbon, and C. Rother. Sphereflow: 6 dof scene flow from rgb-d pairs. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3526–3533, Columbus, OH, USA, 2014.

Y. Hu, R. Song, and Y. Li. Efficient coarse-to-fine patchmatch for large displacement optical flow. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5704–5712, Las Vegas, NV, USA, June/July 2016.

B. Huhle, M. Magnusson, W. Strasser, and A. J. Lilienthal. Registration of colored 3d point clouds with a kernel-based extension to the normal distributions transform. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 4025–4030, May 2008.

M. Isard and J. MacCormick. Dense motion and disparity estimation via loopy belief propagation. In Proceedings of the Asian Conference on Computer Vision, pages 32–41, Hyderabad, India, 2006.

S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, UIST '11, pages 559–568, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0716-1. doi: 10.1145/2047196.2047270. URL http://doi.acm.org/10.1145/2047196.2047270.

G. Izatt, H. Dai, and R. Tedrake. Globally optimal object pose estimation in point clouds with mixed-integer programming. In Proceedings of the International Symposium on Robotics Research, 12 2017.

M. G. Jadidi, L. Gan, S. A. Parkison, J. Li, and R. M. Eustice. Gaussian processes semantic map representation. In RSS Workshop on Spatial-Semantic Representations in Robotics, Cambridge, MA, USA, July 2017.

M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers. A primal-dual framework for real-time dense rgb-d scene flow. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 98–104, Chicago, IL, USA, May 2015.

A. E. Johnson and S. B. Kang. Registration and integration of textured 3d data1. Image and vision computing, 17(2):135–147, 1999.

M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan. Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks? In Proceedings of the IEEE International Conference on Robotics and Automation, pages 1–8, 2017.

C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. In Proceedings of the IEEE International Conference on Robotics and Automation, May 2013.

R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet. Lyft level 5 av dataset 2019. urlhttps://level5.lyft.com/dataset/, 2019.

A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár. Panoptic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9404–9413, 2019.

M. Korn, M. Holzkothen, and J. Pauli. Color supported generalized-icp. In 2014 International Conference on Computer Vision Theory and Applications (VISAPP), volume 3, pages 592–599, Jan 2014.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012a.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages 1097–1105, 2012b.

B. Lee and D. D. Lee. Learning anisotropic ICP (LA-ICP) for robust and efficient 3D registration. In Proc. IEEE Int. Conf. Robot. Automat., pages 5040–5045. IEEE, 2016.

J. Levinson and S. Thrun. Robust vehicle localization in urban environments using probabilistic maps. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 4372–4378, 2010.

J. Levinson, M. Montemerlo, and S. Thrun. Map-based precision vehicle localization in urban environments. In Proceedings of the Robotics: Science & Systems Conference, volume 4, 2007.

T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In European Conf. on Computer Vision, pages 740–755. Springer, 2014.

C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman. SIFT flow: Dense correspondence across different scenes. In Proceedings of the European Conference on Computer Vision, pages 28–42, Marseille, France, October 2008.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3431–3440, 2015a.

J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., pages 3431–3440, 2015b.

B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In Proceedings of the International Joint Conference on Artifical Intelligence, pages 674–679, Vancouver, Canada, August 1981.

A. Makadia, A. Patterson, and K. Daniilidis. Fully automatic registration of 3d point clouds. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, volume 1, pages 1297–1304. IEEE, 2006.

D. Maturana, S. Arora, and S. Scherer. Looking forward: A semantic mapping system for scouting with micro-aerial vehicles. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 6691–6698. IEEE, 2017.

J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet RGB-D: 5M photorealistic images of synthetic indoor trajectories with ground truth. 2016.

J. McCormac, A. Handa, A. J. Davison, and S. Leutenegger. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. In Proc. IEEE Int. Conf. Robot. Automat., pages 4628–4635, May 2017a.

J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet RGB-D: Can 5M synthetic images beat generic ImageNet pre-training on indoor segmentation? In Proceedings of the IEEE International Conference on Computer Vision, 2017b.

F. M. Mirzaei and S. I. Roumeliotis. Globally optimal pose estimation from line correspondences. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 5581–5588, May 2011.

R. M. Murray, Z. Li, S. S. Sastry, and S. S. Sastry. A mathematical introduction to robotic manipulation. CRC press, 1994.

P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from RGBD images. In European Conf. on Computer Vision, 2012.

R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. a. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In IEEE ISMAR. IEEE, October 2011. URL https://www.microsoft.com/en-us/research/publication/kinectfusion-real-time-dense-surface-mapping-and-tracking/.

E. B. Olson. Real-time correlative scan matching. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 4387–4393, May 2009.

S. A. Parkison, L. Gan, M. G. Jadidi, and R. M. Eustice. Semantic iterative closest point through expectation-maximization. In Proceedings of the British Machine Vision Conference, pages 1–17, Newcastle, UK, September 2018.

S. A. Parkison, M. Ghaffari, L. Gan, R. Zhang, A. K. Ushani, and R. M. Eustice. Boosting shape registration algorithms via reproducing kernel hilbert space regularizers. IEEE Robotics and Automation Letters, 4(4):4563–4570, Oct 2019. ISSN 2377-3774. doi: 10.1109/LRA.2019.2932865.

S. A. Parkison, J. M. Walls, R. W. Wolcott, M. Saad, and R. M. Eustice. 2d to 3d line-based registration with unknown associations via mixed-integer programming. In Proceedings of the IEEE International Conference on Robotics and Automation, Paris, France, June 2020.

A. Parra Bustos, T.-J. Chin, and D. Suter. Fast rotation search with stereographic projections for 3d registration. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.

S. Pillai and J. Leonard. Monocular SLAM supported object recognition. In Robotics: Science and Systems, Rome, Italy, July 2015.

F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. Auton. Robot, 34(3):133–148, 2013.

A. Pressley. Elementary Differential Geometry. Springer-Verlag London, 2 edition, 2010.

B. Přibyl, P. Zemčík, and M. Čadík. Camera pose estimation from lines using pl\" ucker coordinates. Proceedings of the British Machine Vision Conference, 2016.

A. Pronobis, F. Riccio, and R. P. Rao. Deep spatial affordance hierarchy: Spatial knowledge representation for planning in large-scale environments. In ICAPS 2017 Workshop on Planning and Robotics, 2017.

C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3D classification and segmentation. arXiv preprint arXiv:1612.00593, 2016.

C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Advances in Neural Information Processing Systems 30, pages 5105–5114. 2017.

C. Rasmussen and C. Williams. Gaussian processes for machine learning, volume 1. MIT press, 2006.

S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In 3-D Digital Imaging and Modeling, 2001. Proceedings. Third International Conference on, pages 145–152. IEEE, 2001.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. International Journal of Computer Vision, 115(3):211–252, Dec 2015. ISSN 1573-1405. doi: 10.1007/s11263-015-0816-y. URL https://doi.org/10.1007/s11263-015-0816-y.

R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In Proc. IEEE Int. Conf. Robot. Automat., Shanghai, China, May 9-13 2011.

R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., pages 1352–1359, 2013.

B. Schölkopf and A. J. Smola. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press, 2002.

B. Schölkopf, R. Herbrich, and A. Smola. A generalized representer theorem. In Computational learning theory, pages 416–426. Springer, 2001.

A. Segal. Generalized-ICP. https://github.com/avsegal/gicp, 2009. [Online; accessed 30-January-2018].

A. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In Robotics: Science and Systems, volume 2, 2009.

J. Servos and S. L. Waslander. Multi-Channel Generalized-ICP: A robust framework for multi-channel scan registration. Robot. Auton. Syst., 87:247–257, 2017.

B. Sevilmis and B. Kimia. Shape-based image correspondence. In E. R. H. Richard C. Wilson and W. A. P. Smith, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 66.1–66.12. BMVA Press, September 2016. ISBN 1-901725-59-6. doi: 10.5244/C.30.66. URL https://dx.doi.org/10.5244/C.30.66.

E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(4):640–651, April 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2572683.

J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In IEEE Conf. on Computer vision and pattern recognition, pages 1–8. IEEE, 2008.

A. Spek, W. H. Li, and T. Drummond. A fast method for computing principal curvatures from range images. arXiv preprint arXiv:1707.00385, 2017.

T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal. Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations. The Int. J. Robot. Res., 31(12):1377–1393, 2012.

J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of rgb-d slam systems. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Oct. 2012.

Z. Sui, Z. Zhou, Z. Zeng, and O. C. Jenkins. Sum: Sequential scene understanding and manipulation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 3281–3288, Sep. 2017. doi: 10.1109/IROS.2017.8206164.

T. Tamaki, M. Abe, B. Raytchev, and K. Kaneda. Softassign and EM-ICP on GPU. In 2010 First International Conference on Networking and Computing, pages 179–183, Nov 2010. doi: 10.1109/IC-NC.2010.60.

M. E. Tipping. Sparse Bayesian learning and the relevance vector machine. Journal of Machine Learning Research, 1(Jun):211–244, 2001.

M. E. Tipping. Bayesian inference: An introduction to principles and practice in machine learning. Lecture notes in computer science, 3176:41–62, 2004.

M. E. Tipping, A. C. Faul, et al. Fast marginal likelihood maximisation for sparse bayesian models. In AISTATS, 2003.

A. K. Ushani, , R. W. Wolcott, J. M. Walls, and R. M. Eustice. A learning approach for real-time temporal scene flow estimation from LIDAR data. In Proceedings of the IEEE International Conference on Robotics and Automation, pages 5666–5673, Singapore, May 2017.

R. Valencia, E. H. Teniente, E. Trulls, and J. Andrade-Cetto. 3D mapping for urban service robots. In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pages 3076–3081, 2009.

S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade. Three-dimensional scene flow. In Proceedings of the IEEE International Conference on Computer Vision, pages 722–729, Kerkyra, Greece, September 1999.

Waymo. Waymo open dataset: An autonomous driving dataset, 2019.

M. Weinmann and B. Jutzi. Geometric point quality assessment for the automated, markerless and robust registration of unordered TLS point clouds. ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences, 2, 2015.

R. W. Wolcott and R. M. Eustice. Visual localization within LIDAR maps for automated urban driving. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 176–183, Chicago, IL, USA, September 2014.

R. W. Wolcott and R. M. Eustice. Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving. The Int. J. Robot. Res., 36:292–319, 3 2017a.

R. W. Wolcott and R. M. Eustice. Robust LIDAR localization using multiresolution Gaussian mixture maps for autonomous driving. International Journal of Robotics Research, 36:292–319, 3 2017b.

Z. Yan and X. Xiang. Scene flow estimation: A survey. arXiv preprint arXiv:1612.02590, 2016.

J. Yang, H. Li, and Y. Jia. Go-icp: Solving 3d registration efficiently and globally optimally. In Proceedings of the IEEE International Conference on Computer Vision, pages 1457–1464, 2013.

F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In ICLR, 2016.

F. Yu, J. Xiao, and T. Funkhouser. Semantic alignment of LiDAR data at city scale. In Proc. IEEE Int. Conf. Computer Vision and Pattern Recog., pages 1722–1731, 2015.

A. Zaganidis, M. Magnusson, T. Duckett, and G. Cielniak. Semantic-assisted 3D normal distributions transform for scan registration in environments with limited structure. In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., Sept. 2017.

A. Zaganidis, L. Sun, T. Duckett, and G. Cielniak. Integrating deep semantic segmentation into 3-d point cloud registration. IEEE Robotics and Automation Letters, 3(4):2942–2949, oct 2018. doi: 10.1109/lra.2018.2848308. URL https://doi.org/10.1109/lra.2018.2848308.

Z. Zeng, Y. Zhou, O. C. Jenkins, and K. Desingh. Semantic mapping with simultaneous object detection and localization. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 911–918, Oct 2018. doi: 10.1109/IROS.2018.8594205.

G. Zhang, J. H. Lee, J. Lim, and I. H. Suh. Building a 3-d line-based map using stereo slam. 31 (6):1364–1377, 2015.

L. Zhang, C. Xu, K.-M. Lee, and R. Koch. Robust and efficient pose estimation from line correspondences. In K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, editors, Computer Vision – ACCV 2012, pages 217–230, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg. ISBN 978-3-642-37431-9.

L. Zhou, C. Xu, P. Koch, and J. J. Corso. Watch what you just said: Image captioning with text-conditional attention. In Proceedings of the on Thematic Workshops of ACM Multimedia 2017, Thematic Workshops '17, pages 305–313, New York, NY, USA, 2017. ACM. ISBN 978-1-4503-5416-5. doi: 10.1145/3126686.3126717. URL http://doi.acm.org/10.1145/3126686.3126717.

X. Zuo, X. Xie, Y. Liu, and G. Huang. Robust visual slam with point and line features. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 1775–1782, Sep. 2017.