Spatial-Semantic 3D Robot Perception with Computational Symmetries

by

Ray Zhang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Robotics)
in the University of Michigan
2024

Doctoral Committee:

Assistant Professor Maani Ghaffari, Co-Chair
Professor Ryan Eustice, Co-Chair
Assistant Professor Nima Fazeli
Professor Jessy Grizzle
Associate Professor Ramanarayan Vasudevan

Ray Zhang

rzh@umich.edu

ORCID iD: 0000-0001-9599-931X

ACKNOWLEDGEMENTS

I express my sincere gratitude to my esteemed advisors, Professor Maani Ghaffari and Professor Ryan Eustice. Maani's guidance during my master's studies, introducing me to my research areas, and instrumental assistance in my admission to the Ph.D. program have been invaluable. Maani has consistently been not only my research advisor but also a friend and a role model. I am deeply inspired by Maani's passion, commitment to novelty, and dedication to writing. The moment I received the Ph.D. offer with Professor Ryan Eustice as my advisor remains etched in my memory. Ryan's confidence and trust in someone he had never met were both humbling and motivating. During my first research internship, Ryan's talk on the technical advancements at TRI provided a firsthand glimpse into what cutting-edge research and leadership in industry entail. Thank you, Ryan, for your unwavering support, encouragement, career advice, and consistently uplifting words. Next, thank you, Professor Jessy Grizzle, for being my collaborator and advisor during the early years of my Ph.D. journey. Jessy's care for his students is truly commendable. I vividly recall his dedicated presence with the team during the sweltering summer days in the Michigan wavefield. I extend my appreciation to my committee members, Professor Nima Fazeli and Professor Ramanarayan Vasudevan, for being willing to be on my committee and provide insightful feedback on my current research and future directions. Special thanks to Denise Edmund for her kindness, helpfulness, and patience. Gratitude is also due to all the Curly lab members who have been integral to my academic journey—Jingwei Song, Minghan Zhu, Sangli Teng, Cynthia Lin, Justin Lin, Bo Fu, Joseph Wilson, Tribhi Kathuria, Junzhe Wu, Tianyi Liu, Xiang Gao, Jinyuan Zhang, Fenghua Yang, and others. Our shared experiences of hard work and enjoyment have forged strong bonds. I am thankful to the early members of the PeRL and Dynamic Locomotion Lab, including Jie Li, Arash Ushani, Josh Mangelson, Steven Parkison, Yukai Gong, Zhenyu Gan, Eva Mungai, Grant Gibson, Oluwami Dosunmu-Ogunbi, Jinsun Liu. As an international student, the warmth and support received from colleagues and friends have been invaluable. I am grateful for the friendships that have been cultivated, and I hope they endure. Last but certainly not least, my deepest appreciation goes to my parents. Your love has been my constant motivation and driving force.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

ABSTRACT

Robot localization and mapping is a process where a robot constructs a global spatial model of a scene based on multiple local observations. It is a fundamental problem supporting other components of robot autonomy, such as planning and navigation. In this thesis, we are concerned with building a robust and semantic-aware framework of robot perception. With the help of modern vision sensors, recent perception techniques are capable of producing fast and accurate estimations for robot positions and 3D geometric structures under feature-rich environments. In perceptually degraded situations, mainstream algorithms' dependencies on feature quality and scene structure are no longer valid. Besides, the advances of deep neural networks provide ways of learning point-wise semantic information. However, these data-driven methods require an enormous amount of labeled data to capture the underlying distribution of the practical real world. Furthermore, effective strategies for fusing the available predictions from upstream machine learning algorithms into a consistent and efficient spatial-semantic environment model remain an ongoing challenge. To address these problems, we propose leveraging the tools from the reproducing kernel Hilbert space and the equivariance learning theory based on the symmetry of input data to construct a continuous and functional representation of the sensor observations. This representation enables a novel formulation of the localization problem that is robust towards sensor noise and outliers. Furthermore, this representation can tightly couple the spatial-semantic relations for different perception tasks.

Specifically, in Chapter 4, we detail the functional and equivariant formulation that integrates geometric and semantic measurements and then introduce a frame-to-frame sensor registration framework. It outperforms the existing frame-to-frame registration methods in accuracy and robustness. Chapter 5 extends the two-frame registration to a multi-frame bundle adjustment formulation. It demonstrates the potential in local and large-scale robot trajectory estimations, with robust performances in feature-less and outlier-rich scenarios. Finally, Chapter 6 presents an unsupervised equivariance learning framework for learning the semantic features of the above sensor representation, which excels in environments with limited ground truth data. It offers robust adaptation and improved resilience against outliers or mismatches in simulated and unseen real-world datasets. The aforementioned contributions enhance the data efficiency, generalization, semantic understanding, and robustness of robot perception techniques within the demanding context of perceptually degraded applications.

CHAPTER 1

Introduction

1.1 Motivation

In recent years, we have witnessed enormous advancements of mobile robots in rural and urban applications. Autonomous systems have the potential to greatly impact industries such as logistics, cargo operations, manufacturing, and energy. They could also play a vital role in emergency response situations like disaster relief, as well as scientific exploration and data collection. In the previous scenarios, robots often operate within previously explored areas to perform routine tasks. In the latter case, robots are required to navigate unknown places and make decisions online. As described in Figure 1, the pivotal question is: At what point will the general public perceive a mobile robot as a genuinely reliable societal and business companion?

For autonomous and human-controlled robots, reliable navigation requires an accurate understanding of their position within their environments. Specifically, robots ascertain their relative poses ”cognitively,” utilizing either a pre-existing mental map of surrounding environments or a newly constructed one. This is rooted in a fundamental problem in classical robot perception literature, Simultaneous Localization and Mapping (SLAM), shown in Figure 3, where the estimation of the robot’s trajectories and environmental models take place concurrently [7]. Over the past decades, remarkable progress has been achieved in various SLAM components, including visual odometry [59], global sensor registration [62, 112, 226, 210], place recognition [90], bundle adjustment [184, 77, 92], etc. With the help of the latest vision and inertia sensors, modern perception techniques have demonstrated rapid and accurate estimations of robot positions and 3D geometric structures [145], provided that sufficient data is available and environments are feature-rich.

However, with the goal of fully autonomous operations, mobile robots will inevitably encounter perceptually degraded situations, such as dust, lousy weather, deserts, forests, low light, and dynamic objects. The vision degradation can cause loss of sensor tracking, resulting in failure to localize and reconstruct novel scenes [200]. For instance, dusty and smoky environments can create numerous false-positive laser reflections [134]. Areas with low light often lack the neces-

Figure 1: The adoption of autonomous robots is increasing at various terrains, utilized for diverse applications [66, 53, 93, 1]. While certain robots concentrate on repetitive tasks within specific and preset environments, others are designed to conquer unexplored and unfamiliar settings. Before mobile robots become commonplace in our life, we need to consider what barriers are still preventing the general public from viewing these robots as safe and dependable partners.

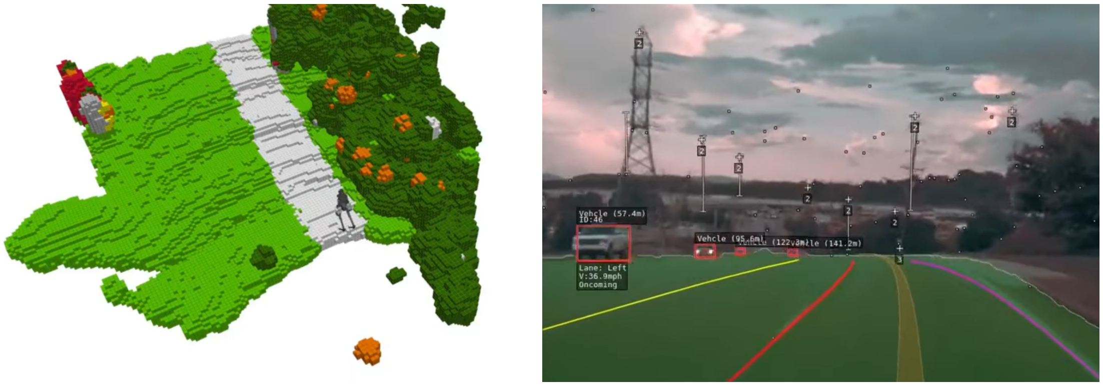


Figure 2: For autonomous operations, mobile robots require incorporating semantic comprehension into their perception systems. Left: The biped robot walks along the traversable sidewalk guided by a semantic 3D map constructed online [66]. Right: The autonomous car navigates along the road while avoiding other cars moving around [4].

sary number of feature points [68]. Repetitive patterns or texture-less regions could disrupt the feature matching process during stereo triangulation [80, 144], illustrated in Figure 4. Forest exploration tasks could experience many recurring patterns in the visual sensors, potentially leading to inaccurate place recognition [67]. All these scenarios could confuse a SLAM system's position tracking and loop closing ability [194]. Some research efforts attempt to represent these outliers or noises through certain probabilistic models [190, 97], while others experiment with learning more expressive and distinguishable features [154, 176].

Besides robustness under challenging environments, a reliable autonomous robot must interact independently with the world, requiring minimal direct human supervision. As shown in Figure 2, an autonomous legged robot needs to comprehend which terrain areas are traversable and which could cause entrapment [66]. The recognition of the locations of different objects, such as

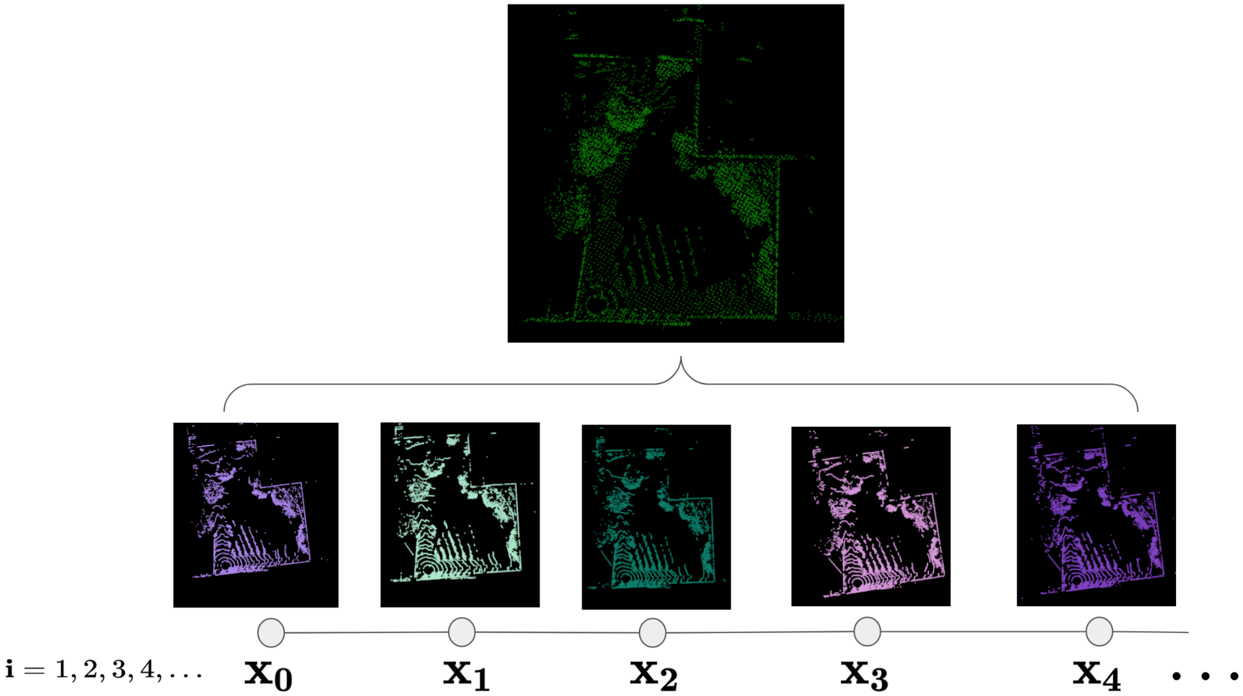


Figure 3: An illustration of a simple SLAM framework consisting of five frames of point cloud observations from laser sensors, collected from the Wave Field [126] in the University of Michigan. The expected outputs include a global world model consistent with each frame's sensor observation and poses at all the timestamps.

sidewalks, trees, and buildings, could aid in its self-awareness of its physical position. Similarly, an autonomous wheeled vehicle needs to discern which parts of the road are obstacle-free lanes and identify dynamic objects to avoid [173, 4]. Should such robots await commands from human operators, potentially leading to several minutes of delay as evidenced by robots like the Mars Rovers [175], they would be compelled to move at an extremely slow pace, thereby increasing the damage risks from themselves or from surrounding moving objects.

Attaining true autonomy thus necessitates integrating scene understanding into the geometric realm [145]. Currently, such semantic knowledge is typically garnered through data-driven methods, e.g., deep neural networks, with substantial volumes of labeled data [49]. Ensuring that the training datasets reflect the actual data distribution in the real world can be difficult. Many efforts have been devoted to constructing a large enough dataset that covers as much as the real world [200, 39], while out-of-the-distribution data could still hit during test time [199]. One example of out-of-training-distribution data emerges from unseen transformations acting on the input domain as shown in Figure 5. Though classical 2D and 3D convolutional neural networks (CNN) excel in encapsulating translational invariance and equivariance of the input data, they encounter challenges around their restricted capacity to detect other types of *symmetries*, such as rotations [36]. Moreover, it remains an open problem how the raw semantic predictions are effi-

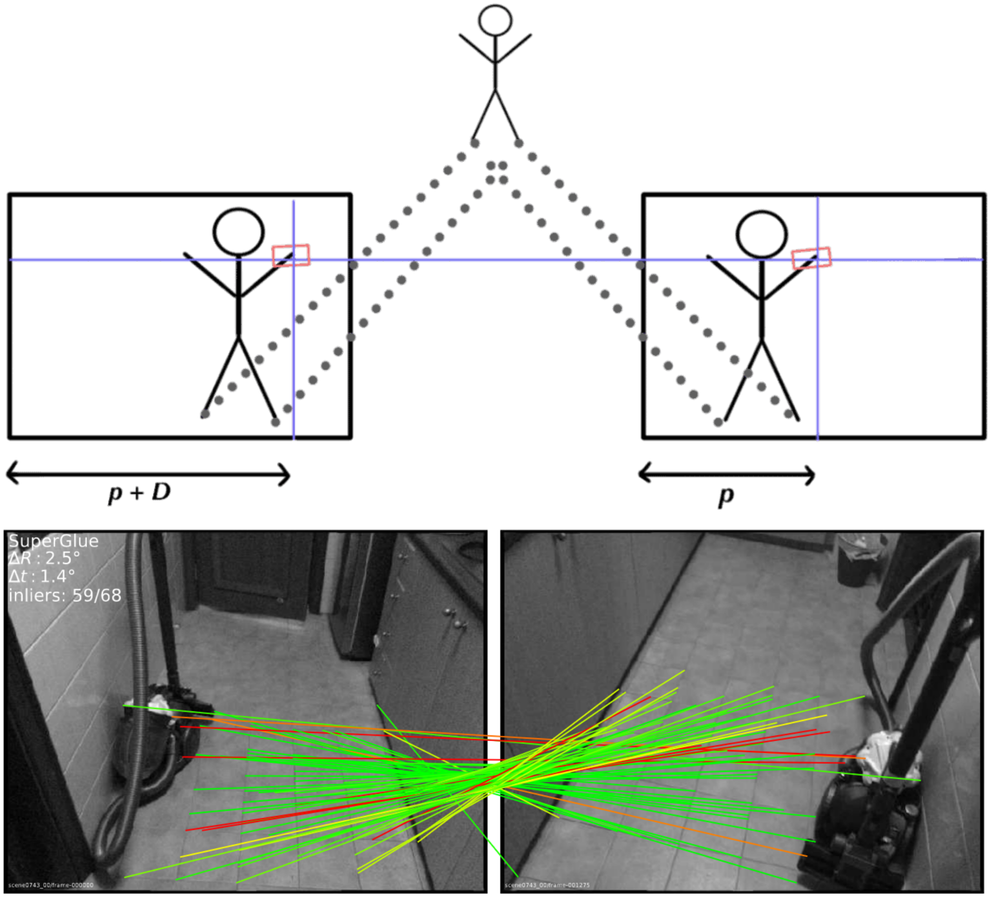


Figure 4: Stereo matching is an essential step in classical SLAM systems. It assumes the presence of sufficiently distinguishable feature points between two sensor observations. Yet, in real-world data, mis-matches are challenging to eliminate. Above: An illustration of how two feature points are matched [203]. Below: Repetitive patterns can confuse the data correspondence process [154].

ciently and thoroughly integrated into the existing perception models [16, 214, 51, 223], so as to simplify the decision-making for downstream planning and control algorithms.

1.2 Research Objectives

The purpose of this thesis is to redefine the robustness and generalization of robot spatial AI frameworks by incorporating semantics knowledge and the inherent symmetric structures of input data through unsupervised algorithms. Our primary focus falls on two essential elements of the SLAM system: *sensor registration* and *bundle adjustment* (BA). Sensor registration addresses the local position-tracking ability of perception systems, while bundle adjustment enhances the long-term global mapping consistency.

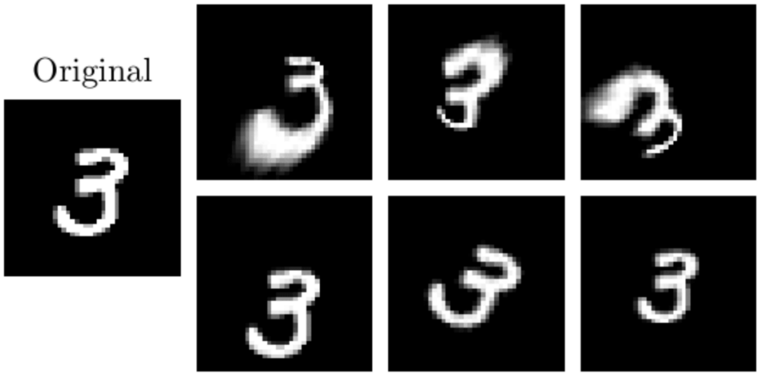


Figure 5: An illustration of how sensor inputs can be impacted with arbitrary transformations [110]. Current datasets may contain some of these transformations [99], but encompassing all possible transformations in the training dataset is unlikely.

In particular, this work develops the following research objectives.

1. A continuous representation of 3D point clouds from camera and LIDAR observations that integrates both geometric structures and semantic understandings.

2. A mathematically well-defined distance metric on such point cloud representation. An optimization procedure robustly implemented for modern hardware compatibility.

3. A deep neural network for sensor registration that respects the natural symmetry of input data and generalizes well from unseen transformed point clouds. A data-efficient way of training the network with limited ground truth labels.

4. A bundle adjustment framework that models the continuous spatial-semantic distribution of a sequence of frames and their topological relationships. Methods of robustly recovering all the frames' poses and the associated map.

1.3 Outline and Contributions

The rest of this thesis is divided into 5 chapters, as summarized below.

Chapter 2 gives a review of the relevant literature. This includes a review of two-frame and multi-frame point cloud registration techniques for robot perception. It also includes a review of local and global bundle adjustment techniques for SLAM frameworks with point cloud inputs.

Chapter 3 covers a semantic map compression method with sparse Bayesian regressions, in particular, with the Relevance Vector Machine. The technique is based on the principle of automatic relevance determination and produces sparse local models that use a small subset of the original dense training set as the dominant basis. The resulting map posterior is continuous, and queries can be made efficiently at any resolution. We evaluate the proposed relevance vector semantic map using two publicly available datasets. The experimental results show that the proposed mapping system has comparable performance with the state-of-the-art techniques while requiring a lower memory footprint.

Chapter 4 presents a novel nonparametric rigid point cloud registration framework, Semantic Continuous Visual Odometry (CVO), that jointly integrates geometric and semantic measurements such as color or semantic labels into the alignment process and does not require explicit data association. The point clouds are represented as nonparametric functions in a reproducible kernel Hilbert space (RKHS). The alignment problem is formulated as maximizing the inner product between two functions, essentially a sum of weighted kernels, each exploiting the local geometric and semantic features. As a result of the continuous models, analytical gradients can be computed, and a local solution can be obtained by optimization over the rigid body transformation group. Besides, we present a new point cloud alignment metric that is intrinsic to the proposed framework and takes into account geometric and semantic information. The evaluations using publicly available stereo and RGB-D datasets show that the proposed method outperforms state-of-the-art outdoor and indoor frame-to-frame registration methods. An open-source GPU implementation is also provided.

Chapter 5 reports a novel Bundle Adjustment (BA) formulation using an RKHS representation called RKHS-BA. The proposed formulation is correspondence-free, enables the BA to use RGB-D and semantic labels in the optimization directly, and provides a generalization for the photometric loss function commonly used in direct methods. RKHS-BA can incorporate appearance and semantic labels within a hierarchical semantic-geometric functional representation that is continuous and does not require optimization via image pyramids. We develop an odometry method using local RKHS-BA graphs that, compared to existing direct odometry methods, RKHS-BA shows highly robust performance in extremely challenging scenes and the best trade-off of generalization and accuracy across extensive experiments.

Chapter 6 presents a novel unsupervised learning framework for correspondence-free 3D point cloud registration, leveraging SE(3)-equivariant networks to accurately recover continuous SE(3) transformations. Traditional methods, often challenged by arbitrary 3D motions and sensor noises in point cloud observations, rely mostly on invariant feature matching or global equivariant feature pooling. Our method diverges by treating point clouds as nonparametric functions in an RKHS, each point described by high-dimensional SE(3) steerable features. The process is structured based on the CVO formulation, which uses 3D coordinates. The minimization in feature space circumvents the need for pairwise point correspondences. The proposed unsupervised inner-outer loop learning strategy excels in environments with limited ground truth data, offering robust adaptation in function space and enhanced resilience against outliers and mismatches. This robustness is evident in various real-world scenarios. The framework's effectiveness is demonstrated through superior performance over existing baselines on the ModelNet simulated dataset and the ETH3D real-world RGB-D dataset.

CHAPTER 2

Literature Review

2.1 Point Cloud Registration of Two Frames

Consider two (finite) collections of points: $X = \{x_i\}, Z = \{z_j\} \subset \mathbb{R}^3$: the point cloud registration process identifies which homogeneous transformation $h \in \mathrm{SE}(3)$ aligns the first point cloud X and the transformed point cloud $hZ = \{hz_j\}$ the "best". In the literature, it is usually formulated as an optimization problem:

$$h^* = \arg \min_{h \in \mathrm{SE}(3)} d(X, hZ) \tag{2.1}$$

where d is a well-defined distance measure between the two point clouds. The objective function (2.1) can be further expressed as correlations between point pairs from the two point clouds. The correspondence can be sparse if each point in the first point cloud is matched to only a single point from the second point cloud. Alternatively, it can be dense, when each point correlates with several points from the second point cloud. This problem can be likened to a *chicken-and-egg* scenario because knowing the optimal pose leads to the understanding of the correct correspondences of the point pairs. Conversely, with the right data association, it is possible to compute the optimal transformation as well.

2.1.1 Classical ICP-based registration

To improve the quality of one-to-one correspondences (*hard* assignment), the work of [30] assumes that only a portion of points can be paired, thus only considering the first few smallest residuals. Many-to-many correspondences (*soft* assignment) are introduced as the weights of the residuals, controlling the "blurriness" of point matches. The weights can come from mutual information [142] or from Gaussian weights [74]. EM-ICP [76] treats the correspondences as hidden variables and uses Expectation Maximization (EM) [13] to infer both the matches and then the transformations.

Point-to-plane [28], plane-to-plane [117], and Generalized-ICP [159] build local geometric structures to the loss formulation. The work of [164] combines multiple Euclidean invariant features. The work of [160] works on an IR camera and uses extra SIFT features from depth images to help keypoint correspondences. The work of [162] uses color/intensity for both association and registration. Color ICP [131] defines a sum of reprojected photometric and depth loss on dense RGB-D point clouds. GICP-RKHS [133] also appends an additional regularizer to the GICP's loss for point intensity via the Relevance Vector Machine [13]. Semantic-ICP [132] treats points' semantic labels and associations as additional hidden variables as a part of the EM-ICP framework.

2.1.2 Mixture of Gaussian-based Registration Frameworks

Probabilistic registration frameworks represent point clouds as discrete [11, 111] or continuous probability densities [91, 19, 33, 56, 61, 83]. Compared to this work, GMM-based methods also use a double sum of Gaussian kernels, combined with the soft data association, but they come from a different theoretical background than the Reproducing Kernel Hilbert Space (RKHS) [8].

Normal Distribution Transform (NDT) defines a discrete collection of bivariate Gaussian distributions to capture local surface structures [10, 111]. Discretization brings automatic soft data association without the need to infer GMM weights. An effective discretization strategy requires suitable voxel sizes and efficient voxel deployment, for instance, a forest of octrees [111], distance-based voxel sizing [114], hierarchical voxel tree deployment [188], and cell clustering [44]. Compared to NDT, the proposed method is also data association-free, but it is further a continuous representation, thus avoiding the above concerns caused by discretization.

Some continuous GMM-based methods minimize the distance between two distributions. Effective distance measures include Jensen-Shannon divergence [191] and the l_2 distance of the two dense [12] or sparse [19] GMMs. Kernel Correlation (KC) [185] maximizes the correlation between two point clouds using M-estimators, in particular, a sum of Gaussian kernels. It has an identical loss function compared to the proposed geometric-only inner product, but its kernel lengthscales stay fixed throughout the optimization. In addition, KC discretizes the space to avoid the quadratic time cost, while our methods remain continuous with the help of GPU parallel computations.

2.1.3 Registration with Invariant Features

Fully connected layers with symmetric operations (max-pooling) in PointNet [139, 5], convolutions of sparse tensors in FCGF [32], sparse bilateral convolution in SPLATNet [168], and graph convolution layers in DGCNN [196] can capture local and global geometric features of point clouds. Examples of utilizing deep geometric features include 3D-Feat-Net [215], PointNetLK [5], etc.

Given extracted features, point correspondences are calculated in the many-to-many [58, 195] or one-to-one way[89, 129]. Correspondences of a point can be interpreted as a probabilistic distribution of its nearby points, predicted by convolutions and softmax operations over those points' feature embeddings [109]. DCP [195] directly multiplies two feature embedding vectors between all point pairs of the two point clouds, followed by softmax operations to get the correspondences. Deep Global Registration [31] adopts convolution layers that take a candidate pair of points $(x, z) = (x_1, x_2, x_3, z_1, z_2, z_3)$ as input, and classifies whether this pair of point lies in a lower-dimensional manifold.

Early works like FPFH [150] create histogram-based local invariant features and are used in global registration. With the advances of deep learning, deep invariant features provide a richer representation that can assist with feature space correspondence search. Encoders such as MLP [139], Graphic Neural Networks [196], KPConv [178] are used for feature extraction that to permutations invariance and local structures. In the correspondence step, direct supervision of inlier and outlier matches is usually required. The method still requires 1-to-1 pairwise matching with either RANSAC or weighted SVD. To make the data association robust, complicated outlier rejection training mechanisms are adopted, assuming enough labeled training data. FCGF [32] uses feature space metric learning with negative mining to filter the outliers by sampling both positive inliers and negative outliers to prevent the features from being biased on the positive samples. Coarse-to-fine strategies in D3FeatNet [6], DCP [195], Cofinet [217], PREDATOR [86], and GeoTransformer [141] enhance match precision by initially focusing on overlapping areas with superpixel matching, followed by finer point correspondences. Particularly, PREDATOR [86] and GeoTransformer [141] leverage Graph Neural Networks and cross-attention mechanisms for feature enhancement and adopt top-k neighbors for associations. These deep learning techniques, integrated with robust optimization methods like those in [210], represent a significant stride in achieving more accurate and reliable point cloud registration.

2.2 Registration of Multiple Frames

Consider a sequence of K frames' robot poses as $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_K\}$ ($\mathbf{T}_I \in \mathrm{SE}(3)$) and the sensor observations $\mathcal{X} = \{X_1, X_2, ..., X_K\}$ at each timestamp: Each sensor observation contains a finite collection of homogeneous points, $X_m = \{\mathbf{x}_1^m, \mathbf{x}_2^m, ... : \mathbf{x}_i^m \in \mathbb{R}^3\}$. The multi-view registration process identifies the transformations that jointly align all the sensor observations into a globally consistent model. It can be formulated as follows:

$$\mathcal{T}^* = \arg\min_{\mathcal{T}} \sum_{(m,n)\in\mathcal{C}} d(\mathbf{T}_m X_m, \mathbf{T}_n X_n) \tag{2.2}$$

where $\mathcal{C}$ is the set of all the frame pairs sharing the views of the global model. When both the shared 3D structure and the frame poses are optimized, it becomes the bundle adjustment (BA) problem [184].

2.2.1 Multi-view Geometric Registration

Point sets registration in direct methods estimates the poses of two or more point clouds to build a single and consistent model [87, 202, 158]. Repeatedly applying frame-to-frame pairwise registration leads to graduate accumulation of drifts because spatial consistency at nearby but non-adjacent frames is not considered. To reduce odometry drifts, some works perform model-to-frame registration, which fuses several latest point clouds into a local map with previous pose estimations, then registers the latest frame with the map [202, 201]. Model-to-frame registration requires accurate localization in earlier frames; otherwise, it risks yielding an inconsistent map as the registering source.

On the other hand, jointly estimating the poses of multiple point clouds can evenly distribute the errors and demonstrate accurate registration results in real datasets, but it requires the Expectation-Maximization (EM) procedure to infer data correspondence across multiple frames [192, 75, 43, 115].

2.2.2 Photometric Bundle Adjustment

Direct BA methods take wrapped photometric residuals from a large number of image pixels [125, 60, 59, 202]. Keyframe-based direct methods [96, 60, 158] usually construct residuals by projecting one frame's intensity image to another. Map-centric methods [125, 202, 42] project the map elements onto the image pixels and establish the photometric loss. To improve robustness against outliers, robust estimators like T-distribution [13] and Huber-loss functions are wrapped around intensity residuals [96, 59].

A class of hybrid methods still uses dense or semi-dense points without relying on photometric losses. For example, SVO [63] performs feature alignment after dense tracking and converts the problem into classical feature-based solvers. Voldor [116] models the dense optical flow residual distribution with an empirical Fisk residual model.

2.2.3 Classical and Semantic Data Association

Data association is the process of registering features observed in different frames. It builds the topology of the factor graph and is heavily discussed in community [18]. Existing feature-based backend solvers like g2o [77] and iSAM2 [92] assume known data association hypotheses from the frontends. These hypotheses can come from matching invariant visual feature points [108, 148] with methods like optical flow tracking [156] or stereo feature matching [165, 80]. In comparison, direct backends usually adopt projective data association. Keyframe-based direct methods [59, 60] project the points onto other frames' epipolar lines and search for the pixel groups with the minimum intensity. Map-centric methods [202] project the map elements onto the image pixels and establish the correspondence. They differ from feature-based systems in the ability to adjust associations during optimization.

The robustness of backends against incorrect data association hypotheses can be enhanced by considering the associations as latent variables. [16]. One strategy is adding weights as additional variables to the potential data association hypothesis and optimizing both the poses and the weights [171, 2]. Another approach is the use of Non-Gaussian mixture models, for example, max-mixtures, to model multiple uncertain data association hypotheses [127, 50, 51], and can be solved by methods like nonparametric Bayesian belief propagation [64] or Dirichlet process [118, 221].

2.2.4 Learning-based Bundle Adjustment

Recent works introduce deep neural networks' predictions into the bundle adjustments of multiple frames. One class of works aims to utilize accurate monocular depth estimations and pixel associations from neural networks and then perform classical or differentiable BA. Zhou [228] uses an end-to-end learning approach with supervision from view synthesis. It has a mono-depth network and a multi-view pose estimation network that concatenates multiple images directly. It assumes 1) the scene is static without moving objects; 2) there is no occlusion/disocclusion/illumination variance between the target view and the source views; 3) the surface is Lambertian, so the photo-consistency error is meaningful CNN-SLAM [174] generates depth from CNN and then relies on a photometric-residual-based classical non-differentiable BA to optimize the poses. DVSO [212], TANDEM [100] and D3VO [211] use Monocular deep depth networks to initialize the stereo disparity values. The loss function of the odometry includes the reprojected pixel difference residuals(self-supervised). Besides initialization, the depth map estimated from BA is further projected back to the right image to generate a disparity map and compare it with the network's disparity prediction as another loss term. BA-Net [172] makes the BA process differentiable where steps like the damping factor are predicted. RAFT [176] and DROID-SLAM [177] use RNN to predict dense flow matches and then use a supervised dense photometric BA.

Another class of learning-based BA methods uses implicit neural embeddings as frame representations instead of predicting depth and pixel matches directly. CodeSLAM [14] and DeepFactors [41] use a deep compact code that encodes geometric information of each keyframe image. The depth can be inferred from the encoding and the intensity image with linear combinations of codes from multiple frames. The poses and the code are optimized together. iMap [169] provides a real-time implicit map representation via MLP. The network is initialized randomly. To train the MLP with depth, iMap queries along the ray in discrete depths for every pixel and compares it with the network's predictions. The resulting depth prediction is a weighted sum of depth value predictions along the ray. Nerf-SLAM [146] uses Nerf as the scene representation and can generate photorealistic maps.

2.2.5 Lidar Bundle Adjustment

Unlike camera sensors, Lidar observations have significantly finer depth measurements but are natively sparse; thus, strict point-wise correspondence might not exist. Point-to-plane ICP [117] and GICP [159] choose a specific geometric entity, planes, centered at each point and minimize the point to the closest-associated plane's Euclidean or Mahanobis distances. For better odometry, the point-to-plane distance loss can be extended to multi-frame local bundle adjustment algorithms. For instance, LOAM [220] minimizes both point-to-plane and point-to-line distances. The associations of points to planes and lines are computed via geometric distances. LegoLOAM [163] improves association from 3D distance to 2D Lidar range images. Suma further tries surfel maps and projective association. MULLS [130] further proposes using different hand-crafted feature points, including information on linear, vertex, and planar points. Its tracking part uses data association with feature points, and the loss functions consist of point-to-point, point-to-plane, and point-to-line distances. In the backend part, the map consists of submaps. Inter-submap optimization comes first, then intra-submap optimization runs second. Lastly, a PGO is added to optimize the global trajectory.

Global lidar bundle adjustment adds the loop closure constraint to the procedure. Unlike pose graph optimization (PGO), which only considers the pose data's consistency, the lidar bundle adjustment tries to simultaneously optimize the full global map consistency of all the frames. GICP Cost Factor [101] group all the frames into 10-20 submaps, each contained in voxels for fast query during data association. The full costs include the submap overlap loss, and a loop closure pose constraint loss. BALM [106] constructs the map consisting of adaptively-sized voxels of features, such as edges and planes. The loss function contains the point-to-plane and point-to-edge distances. The original point-to-plane minimization problem is transformed into minimizing the eigenvalue of points covariance in each voxel. HBA [105] uses a hierarchical submapping strategy to boost the running efficiency of BALM. CT-ICP [47] proposes a loop closure procedure aggregating point clouds projected onto a 2D elevation image.

2.3 Equivariant Learning in Registration

Equivariance is form of symmetry for a map such that given a transformation in the input, the output changes in a predictable way determined by the input transformation [36]. A function $f : X \to Y$ is *equivariant* to a set of transformations G, if for any $g \in G$,

$$g_Y f(x) = f(g_X x), \forall x \in X \tag{2.3}$$

For example, applying a translation on a 2D image and then a convolution layer is identical to processing the original image with a convolution layer and then shifting the output feature map; hence, convolution layers are translation-equivariant [36]. Other group convolution works include discretizing rotations into finite groups like the Dihedral group and continuous sampling with Monte-Carlo [110]. For 3D data, Cohen's icosahedron convolution theory [35] and applications such as EPN [23] and E2PN [229] leverage finite group discretization in point cloud analysis. These methods efficiently encode features across various SO(3) angles. In addition, to learn translation-equivariant features, they incorporate traditional convolution techniques.

Another approach involves continuous steerable feature maps in higher-order group representations, demanding significant computational resources for calculating coefficients [179, 37, 65]. VectorNeuron [48] offers a more computationally efficient solution using only type-1 features. Existing equivariant methods with continuous group representations are mainly applied in physics and chemistry, where their performance in real robotics data is under test.

2.4 Semantic Mapping

2.4.1 2D Segmentation Methods

Advances in deep CNN for 2D semantic segmentation provide sufficient accuracy [107, 25, 153, 57, 216] in both indoor and outdoor datasets [79, 123, 40]. In [15], the original point cloud is projected onto multiple camera views, each segmented with CNN, and then back-projected to the 3D space. The label of a single point is decided by the class of the nearest back-projected 2D pixel.

2.4.2 3D raw point cloud segmentation methods

[139] proposes PointNet that combines spatial feature blocks of points into a point-wise representation. [140] extends PointNet into large-scale scenes by aggregating blocks of multiple scales and sharing features of neighboring points. Similarly, [140] exploits local structures of different scales by performing hierarchical sampling and grouping nearby points. SqueezeSeg in [204] also processes raw point clouds, but instead of taking them as unstructured sets, SqueezeSeg spherically projects all points onto a dense 2D image, which will be the input of neural nets. Furthermore, [102] introduces attributed directed graphs as a form of compact point-group representations, which can capture the contextual relations between neighboring objects. [138] leverages 2D object detectors to find regions of interest in the projected 3D point cloud and then assigns labels to the frustum of each region using PointNet[139].

2.4.3 3D Voxel-Based Semantic Map Fusion

After obtaining priors in local frames, fusing the predictions of nearby points from multiple frames produces a global semantic map. Early work in dense semantic mapping back-projected labels from a segmented (2D) image to the reconstructed 3D points and then assigned each voxel or mesh face to the most frequent label according to a label histogram [81, 161]. Bayesian frameworks have been utilized to fuse labels from multiple views into a voxel-based 3D map. In [166], probabilistic segmentation of multiple (2D) images obtained by *Random decision Forests* (RFs) was transferred into 3D and updated using a Bayes filter. [208] proposed DA-RNN, which integrates a deep network for RGB-D video labeling into a dense 3D reconstruction framework built by KinectFusion [88]. DA-RNN yields consistent semantic labeling of indoor 3D scenes; however, it is assumed that semantic labels and geometric information are independent, and therefore, the consistency largely depends on the performance of data association computed by KinectFusion. The 3D label fusion is done by updating a probability vector of semantic classes and choosing the label with the maximum probability. [170] leverages recurrent neural cells to fuse the label updates.

RVSM also does not directly fuse the prior distributions into the voxel map. Instead, it compresses the priors into a sparse set of relevance vectors by training local RVMs. Then, query results from an RVM are fused into the voxel map using Bayesian updates similar to [213].

2.4.4 Semantic Mapping Refinement

In semantic mapping, structure prediction methods such as CRFs are used to improve the map consistency in the 3D space. In [98], a voxel-CRF model is proposed to capture the geometric and semantic correlation by constructing a CRF over the 3D volume. In [82], a Kalman filter is used to transfer 2D class probabilities obtained by RFs to the 3D model; then, 3D labels are further refined through a dense pairwise CRF over the point cloud. In [225], a higher-order CRF model is used to enforce temporal label consistency by generating higher-order *cliques* from *superpixels* correspondences in an RGB-D video, which improves the precision of the semantic maps.

A common feature of the works mentioned earlier is the discretization of the space prior to map inference. That is, once the map is inferred, the prediction cannot be computed at an arbitrary point. In this work, we propose an alternative solution for the problem of dense 3D map building that is continuous and, at the same time, sparse. RVSM incrementally learns relevance vectors which are dominant basis vectors in the data, and builds a sparse Bayesian model [182] of the 3D map. As a result, the semantic information is compressed into compact relevance vectors, and queries can be made efficiently at any desired location.

CHAPTER 3

Compressed 3D Semantic Map Inference via Relevance Vector Machine

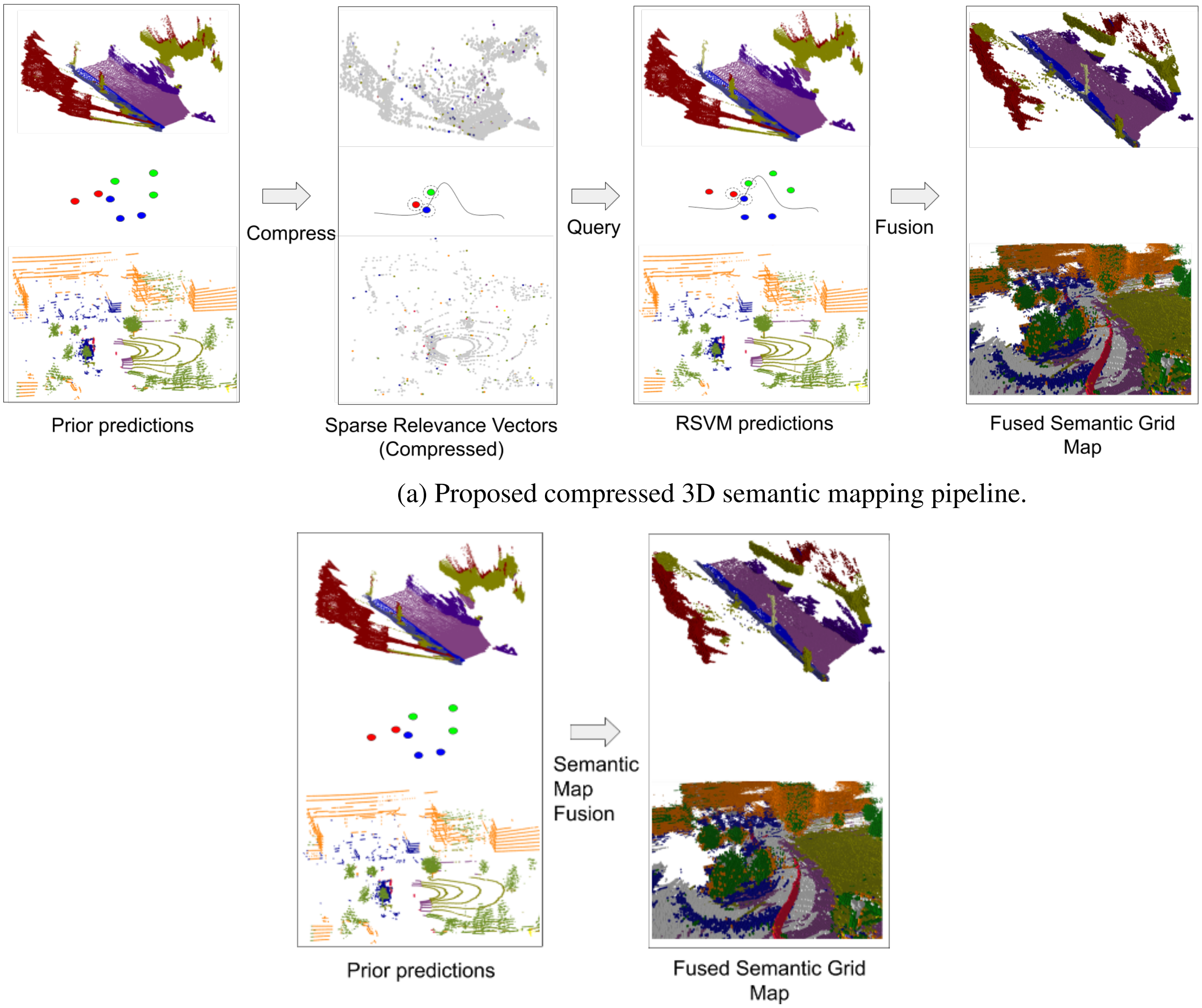
3.1 Overview

Dense robotic mapping is a challenging, high-dimensional, sequential inference problem. This chapter reports on the development of a compressed 3D semantic map inference pipeline through sparse Bayesian models, in particular, the relevance vector machine. The technique is based on the principle of automatic relevance determination and produces sparse local models that use a small subset of the original dense training set as the dominant basis. The resulting map posterior is continuous, and queries can be made efficiently at any resolution. We evaluate the proposed relevance vector semantic map using two publicly available datasets. The experimental results show that the proposed mapping system has comparable performance with the state-of-the-art techniques while requiring a lower memory footprint

3.2 Introduction

Enriching geometric maps with semantic knowledge improves the scene understanding, navigation, and interaction abilities of robots. Current mobile robots rely on three-dimensional (3D) maps for these ends. A 3D semantic mapping system requires it to be not only accurate but also memory and computationally efficient so that it can be used in real robotic applications. In addition, the system needs to be scalable for long-term mapping experiments; that is to say, the mapping system has the ability to reconstruct large maps on demand. In particular, the latter feature is a necessity as the drift caused by odometry and tracking algorithms is often rectified by *loop-closures* within Simultaneous Localization And Mapping (SLAM) systems, resulting in discrete changes in the trajectory of the robot.

The most common approach to build a semantic map is using a voxelized representation of 3D



(a) Proposed compressed 3D semantic mapping pipeline.

(b) Pipeline proposed by [213].

Figure 6: The semantic mapping and fusion pipeline of RVSM, and a comparison with the baseline[213]. Although the proposed system shares similar properties of the baseline, such as final voxel map representation and Bayesian updates per semantic class, it consists of an extra sparse model-building step that allows for map compression and continuous queries at any desired locations.

space and *undirected graphical models* (such as *Markov random fields*) or *Conditional Random Fields* (CRFs) to model the conditional independence between voxels [122]. CRFs are *discriminative* models that are suitable for spatial inference. However, a primary drawback of these approaches is that the resulting representation is discrete from the beginning. The map's resolution is often fixed, and once the map has been inferred from raw sensory data, its resolution cannot be increased. Furthermore, although this approach produces satisfactory local maps, to store a semantic map, all voxels have to store the label distribution, which requires a substantial amount of memory and is not scalable.

In this thesis, we propose a semantic robotic mapping technique, shown in Figure 6, that can *compress* dense semantic information of the scene via *sparse Bayesian models* [182], in particular, the *Relevance Vector Machine* (RVM), without or with minor compromise on prediction performance. While CRFs perform inference on fixed voxels after the 3D space is discretized, RVMs are able to perform high-dimensional sequential inference directly on point clouds with prior distributions. After a Relevance Vector Semantic Map (RVSM) is trained, we can uncompress the semantic information by performing efficient *queries* at any desired location. For application, RVSMs can be efficiently fused into a semantic voxel map using Bayesian updates. In summary, the main contributions of the present work are as follows:

1. We propose a compressed and continuous semantic map representation and integrate it within a pipeline to construct semantic occupancy grid maps.

2. The open source implementation of the proposed system is available at: https://bitbucket.org/perl-sw/rvsm/.

3. We provide experimental results for evaluating the performance of the proposed mapping technique using two publicly available datasets, namely, KITTI [69] and NCLT [22].

The remainder of the chapter, after reviewing the related work in the following section, is organized as follows. Section 3.3 explains the problem setup via sparse Bayesian modeling. Section 3.4 shows the proposed pipeline for compressed 3D semantic map inference. The experimental results using two publicly available datasets are presented in Section 3.5. Finally, Section 3.6 concludes the chapter and provides future work directions.

3.3 Problem Setup

In this section, we formulate the semantic mapping problem and introduce a sequential sparse Bayesian learning algorithm to build a semantic map incrementally. Let $\mathcal{X} \subset \mathbb{R}^3$ be the set of spatial coordinates for map inference, and let $\mathcal{C} = \{1, 2, 3, ..., n_c\}$ be the set of semantic class labels. A subset $\mathcal{Z} \subset \mathcal{X} \times \mathcal{C}$ is the set of possible observations, where an observation is a spatial coordinate associated with a semantic label. A set of existing observations consists of an n_t-tuple random variable $(Z_1, \dots, Z_{n_t})$ whose elements can take values $z_i \in \mathcal{Z}$, $i \in \{1\colon n_z\}$ where $z_i = [x_i, t_i]^\mathsf{T}$, $x_i \in \mathcal{X}$, and $t_i \in \mathcal{C}$. The training set is a subset of the observations, denoted by $\mathcal{D} \subseteq \mathcal{Z}$, where $n_t \leq n_z$ is the number of training points. Let $\mathcal{M}$ be the set of possible assignments of semantic maps, that is, all random variables defined on $\mathcal{X}$ with values in $\mathcal{C}$. A semantic map

$m \in \mathcal{M}$ of the environment associates to each point x_i in $\mathcal{X}$ a probability distribution over $\mathcal{C}$. Given observations $z = (z_1, .., z_{n_z})$, we wish to estimate $p(M = m \mid Z = z)$.

3.3.1 Point-wise Classification via Relevance Vector Machine

To simplify the problem, we use one binary RVM classifier for each class and adopt a *one-vs.-rest* approach for building a n_c-class classifier. Let $x \in \mathcal{X}$ be a 3D input vector before taking the spatial coordinates of the raw point cloud. Given the existing observations $\mathcal{D} = \{(x_i, t_i) : t_i \in \mathcal{C}\}_{i=1}^{n_t}$, for any new input x, the estimate of x's posterior probability of class membership for each class $c_k \in \mathcal{C}$, can be computed from the logistic sigmoid function as follow.

$$p(c_k = +1|\mathcal{D}, x) = \sigma(y(x; w)) = \sigma(\sum_{j=1}^{n_b} w_j \phi_j(x)) = \sigma(\phi(x)w), \tag{3.1}$$

where $y(x; w)$ is a linear model (linear in weights w) to approximate the *latent* function of interest, and $\phi(x) := [\phi_1(x), \ldots, \phi_{n_b}(x)]$ is the kernel basis feature vector. The basis vector $\phi_j(x)$ is replaced with a vector of kernel functions $(1, k(x_1, x_j) \ldots, k(x_{n_b}, x_j)$, where $k(\cdot, \cdot)$ evaluates the similarity between two inputs by implicitly mapping them to a *feature space* [157].

The objective is to learn an optimized weight vector w as well as the number of basis vectors n_b, such that it generalizes well to new inputs x_* (test data). When a significant portion of weights becomes small or zero, the linear model becomes sparse [21].

For better generalization, regularization in the form of zero-mean Gaussian Distribution on weights is added to encourage the target function to be smooth and infer which input vectors are relevant [124]. Instead of computing a *point estimate* for the weights, a Bayesian predictive framework infers the *posterior distribution* over w via Bayes' rule:

$$p(w|t, \alpha) = \frac{p(t|w)p(w|\alpha)}{p(t|\alpha)}. \tag{3.2}$$

Here $p(t|w)$ is the *likelihood* of the observations, in the form of a Bernoulli likelihood

$$p(t|w) = \prod_{i=1}^{n_t} \sigma(y(x_i; w))^{t_i} [1 - \sigma(y(x_i; w))]^{1-t_i}, \tag{3.3}$$

$p(t|\alpha) = \int p(t|w)p(w|\alpha)\mathrm{d}w$ is the *marginal likelihood.* $p(w|\alpha)$ is the zero mean Gaussian prior defined over w using the principle of ARD [124],

$$p(w|\alpha) = (2\pi)^{-n_b/2} \prod_{j=1}^{n_b} \alpha_j^{1/2} \exp(-\frac{\alpha_j w_j^2}{2}), \tag{3.4}$$

with the inverse variance *hyperparameter* $\alpha := [\alpha_1, \ldots, \alpha_{n_b}]^\mathsf{T}$. Each α_j individually controls the effect of the prior over its associated weight w_j. During inference, many of these hyperparameters approach infinity; the posterior distributions of their associated weights peak around zero. Consequently, only a few basis functions with non-zero weights survive in the final classification model, resulting in a sparse model. The basis functions that survive are called *relevance vectors* [181], forming a sparse representation of the semantic map.

The most-probable weights $w^\star$ can be estimated iteratively with an approximate inference strategy [182]:

$$w^\star = \arg\max_{w} \sum_{j=1}^{n_t} [t_j \log y_j + (1 - t_j)\log(1 - y_j)] - \frac{1}{2} w^\mathsf{T} A w, \tag{3.5}$$

where $A := \mathrm{diag}(\alpha_1, \ldots, \alpha_{n_b})$ and $y_j := \sigma(y(x_j; w))$ is the prediction on the j-th observation point cloud. Then the Laplace approximation method provides a local Gaussian approximation of the weight posterior around $w^\star \sim \mathcal{N}(\mu, \Sigma)$:

$$\mu = A^{-1}\Phi^\mathsf{T}(t - y), \quad \Sigma = (\Phi^\mathsf{T} B \Phi + A)^{-1}, \tag{3.6}$$

where $\Phi_{ni} := \phi_i(x_n)$ and $B := \mathrm{diag}(\beta_1, \ldots, \beta_{n_t})$, with $\beta_i := \sigma(y_i)[1 - \sigma(y_i)]$.

3.4 Compressed 3D Semantic Mapping Pipeline

As shown in Figure 6, the compressed semantic map inference system comprises three steps: `compress()`, `query()` and `fusion()`. In the following, we describe each step briefly.

- `compress()`: As described in Section 3.3, (3.5), and (3.6), `compress()` takes prior semantic predictions for the point cloud as the input, and outputs the sparse relevance vectors for each class as the compressed semantic map representation. The prior semantic predictions as existing observations either come from back-projecting 2D image segmentation to 3D or raw 3D point cloud segmentation. These observations are used to train the internal relevance vector machine for each frame and infer the weight parameters of the corresponding classification model. The resulting sparse relevance vectors store the semantic information of the entire map.

- `query()`: In order to generate a semantic voxel map with a certain resolution, we query at occupied locations in an occupancy grid map using the classification prediction model in (3.1) obtained from `compress()`. Thus, the input of this stage can be the centroids of the occupied voxels (if we discard all the priors to reduce memory consumption) or simply the points in the previous observations (for maximum possible accuracy). Given the spatial coordinates of those query points, we obtain the probability of each class. At the end of this stage, we can already build a posterior semantic point cloud map.

- `fusion()`: With the resolution chosen as above, we can fuse the prediction results from `query()` for sequential frames using grid mapping systems such as Octomap[84] and scrolling occupancy grids of [113]. To integrate semantic information into the grid, we store the distribution for each class in all the voxels and adopt a recursive counted averaging label update. A voxel's probability distribution of semantic labels is obtained as follows: Given n points that already fall into the voxel, with its current label distribution being $\mathbf{p}^{(n)}$, when a new point with label distribution of $\mathbf{p}_{\text{new}}$ falls into this voxel, the updated distribution for the voxel can be computed as $\mathbf{p}^{(n+1)} \propto n\mathbf{p}^{(n)} + \mathbf{p}_{\text{new}}$. Finally, the color of each voxel is decided by the label with maximum probability.

3.5 Results and Discussion

We evaluate the semantic prediction performance and multi-resolution query ability of the proposed semantic mapping method with two publicly available datasets, dense point cloud from the stereo camera on KITTI [69] and sparse point cloud from the laser scans on NCLT [22].

3.5.1 Evaluation Metrics

To evaluate the semantic prediction accuracy of RVSM and compare with the state-of-the-art semantic mapping systems, three metrics are used to quantify the results, *Recall (Sensitivity)*= $\frac{\text{TP}}{\text{TP+FN}}$, *Precision*= $\frac{\text{TP}}{\text{TP+FP}}$, *Mean Intersection over Union (mIoU)*= $\frac{\text{TP}}{\text{TP+FN+FP}}$. We calculate these metrics for each semantic class and report the class average results.

3.5.2 KITTI Dataset

3.5.2.1 Experimental Setup

The KITTI odometry dataset consists of 22 outdoor-scene stereo sequences recorded by a sensor mounted on a driving vehicle. We evaluate our method on 25 test images from sequence 15 with labeled data [161].

We choose a CRF-optimized 3D semantic occupancy mapping system [213] as our baseline and quantitatively compare our results with it. The baseline back-projects a 2D convolutional neural network segmentation to 3D as the prior prediction and directly fuses the prior from multiple views

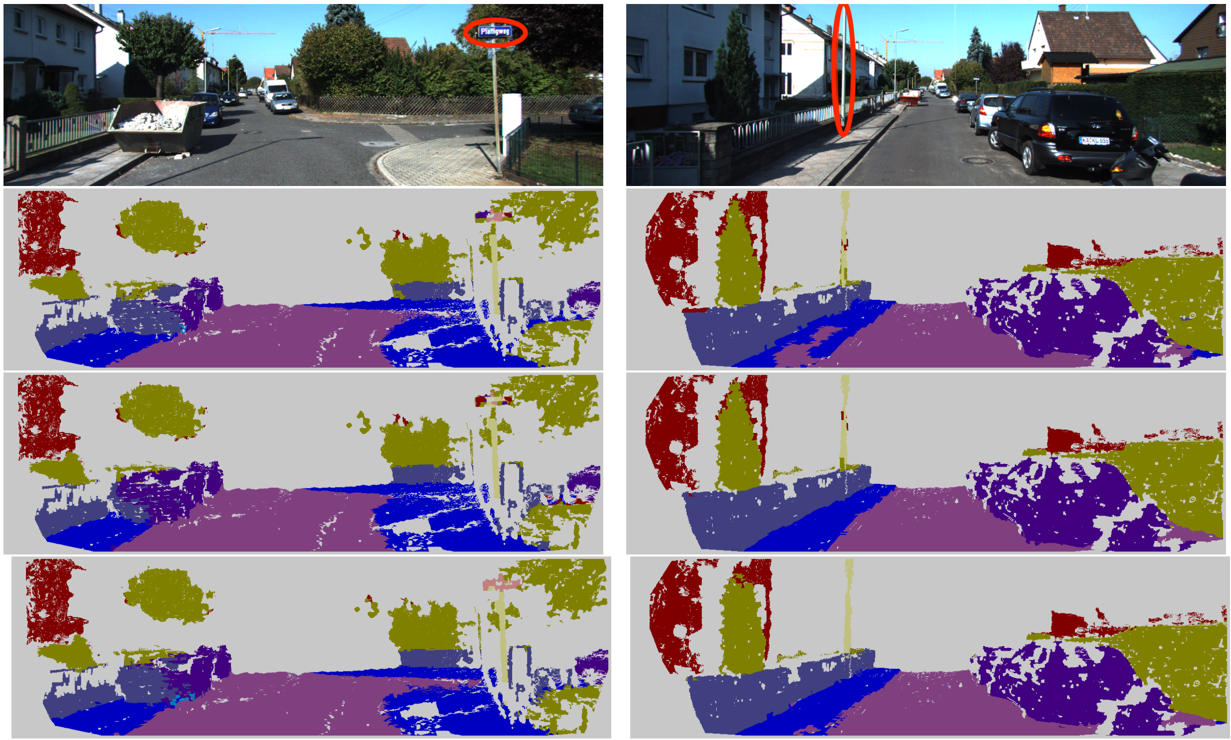


Figure 7: Example on KITTI Sequence 15 after projecting the occupied voxels from the semantically fused grid map to the camera image plane: Up to down, respectively: RGB image, prior result, CRF [213] result, RVSM result. RVSM is able to capture the traffic sign on the left image and the pole on the right image.

into a scrolling occupancy grid map [113] by a Bayesian update procedure. After the grid map is built, higher-order CRFs are utilized to optimize each grid's label with the assumption that the labels of voxels corresponding to a superpixel are consistent.

In our comparison, we adopt the same data pre-processing as the baseline. We first use LIBELAS [70] to generate dense and accurate depth maps from rectified graylevel stereo pairs. By projecting depth maps to 3D space and discarding points more than $40m$ away, we obtain dense point clouds as the training data for RVSM. To obtain the prior, we use the same network as adopted by the baseline, Dilated CNN [216], which is trained on KITTI data from another sequence, and back-project the 2D segmentation to 3D. After the relevance vector machine is trained for each frame, we attach each relevance vector machine to a pose obtained using ORBSLAM2 Stereo [121]. At this point, we have the compressed and continuous semantic map representation RVSM in the global coordinate frame. For evaluation, we uncompress the RVSM by fusing the query points into the scrolling occupancy grid map. As described in Section 3.4, queries can be done at any occupied location with any resolution.

We compare the prediction results for 7 common semantic classes, i.e., *building, vegetation, car, road, fence, sidewalk, pole*. As our baseline, we project all labeled grids onto the camera image

Table 3.1: KITTI Experiment RVM Model Selection: We run sequential RVM inference for 500 iterations per class on 0.01 of the training data. The current kernel and hyperparameters are chosen using sample data and training a Gaussian process [143].

Kernel	Length-Scales (x, y, z)	Signal Variance	Iterations	Downsample Rate
Matern ($\nu = 5$) [143]	$(1.3452, 1.2746, 1.9771)$ m	2.2133	500	0.01

plane, ignoring grids that are $40m$ or more from the camera, and evaluate them on 2D ground truth labels. Sky labels are discarded when evaluating due to the inaccurate depth values. RVM model selection in this experiment is described in Table 3.1.

3.5.2.2 Qualitative Results

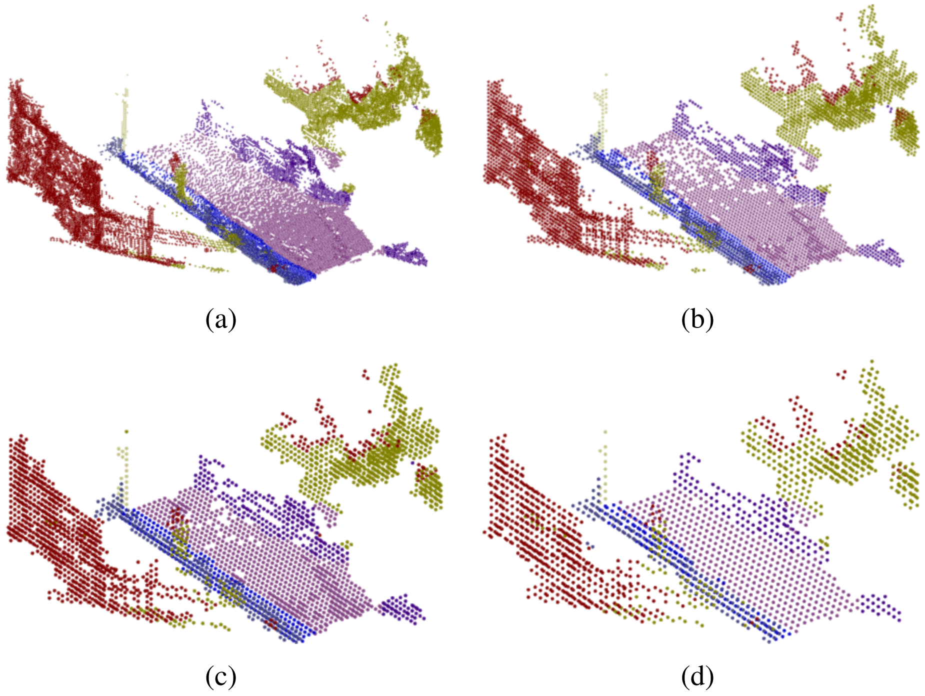


Figure 8: Query for different resolution. From (a)-(d): map resolution changes from 0.1 m to 0.4 m.

We demonstrate the advantage of RVSM over the semantic prior map (directly transferring labels from 2D to 3D) and a semantic map optimized by CRF [213]. Figure 7 is generated by projecting the grid map into the camera image plane. Due to the small size of the sign and the color similarity of the pole and building in the raw RGB images, the prior map is unable to discriminate

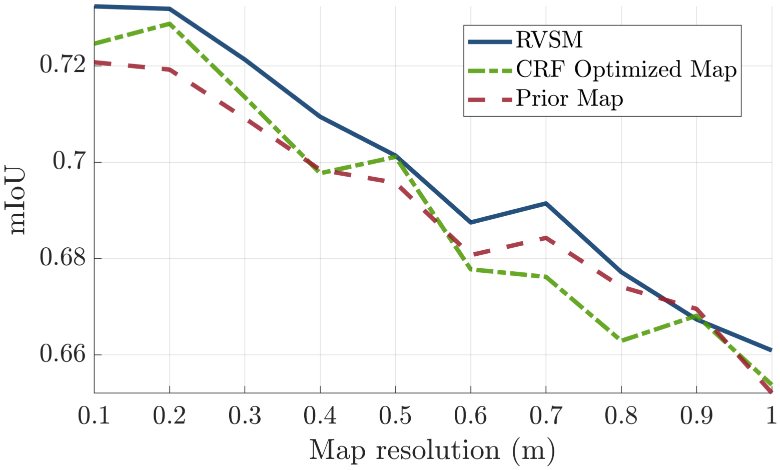


Figure 9: mIoU at different resolutions for Prior, CRF, and RVSM. RVSM only does one query on the prior points' spatial coordinates and generates the map for all the resolutions. Here the resolution means the size of the voxel. When the resolution decreases, more points (possibly with different labels) fall into one voxel, resulting in a lower mIoU value. The larger the voxels are, the more information is lost from CRF's discretization. On the contrary, RVSM performs compression before voxel map fusion, which preserves local semantic consistency better than post-processing.

them from the background. The CRF-optimized map is also unable to recover the correct labels completely due to inaccurate superpixel segmentation when the objects become small in the input images. RVSM, on the other hand, which builds the inference model before fusing the semantics and discretizing the space, successfully recovers the correct labels for even detailed objects.

We also show the continuity of RVSM in Figure 8 by querying voxel centroids of grid maps with multiple resolutions. We build the RVSM from prior predictions for one frame. After the RVSM is built, the semantic information of the scene is compressed in the format of several relevance vectors. Due to RVSM's continuity property, we can build grid maps afterward with any desired resolution, depending on the applications.

3.5.2.3 Quantitave Results

In this section, we quantitatively evaluate the proposed technique in terms of overall performance, multi-resolution performance, and compression ability, as well as compare it with other approaches.

Comparison on overall performance: We compare the overall semantic prediction performance with the prior map and the CRF map [213] for each class using three metrics: recall, precision, and IoU. We build three grid maps for the first 100 frames in sequence 15 and project the grids into 25 images with ground truth for evaluation, discarding points that are further than $40m$ away

Table 3.2: Quantitative results on KITTI odometry sequence 15 test set [161] for 7 semantic classes, containing 25 ground truth images. The query points of RVSM are chosen to be at all the observed prior locations.

Metric	Method	Building	Vege.	Car	Road	Fence	Sidewalk	Pole	Average
Recall	Prior Map	98.2	**99.2**	98.6	91.4	**96.3**	82.8	58.2	89.2
	CRF Map[213]	98.2	98.5	**98.7**	91.6	96.0	**86.0**	58.0	89.6
	RVSM	97.7	98.9	98.2	**94.5**	96.0	85.0	**70.0**	**91.4**
Precision	Prior Map	86.5	88.9	**81.0**	74.5	**76.5**	76.7	**58.0**	77.4
	CRF Map[213]	88.3	**91.4**	80.5	76.1	73.0	76.8	56.7	**77.6**
	RVSM	**88.4**	90.6	80.6	**77.1**	75.1	**77.2**	51.6	77.2
IoU	Prior Map	85.2	88.3	**80.0**	69.6	**74.3**	66.2	40.9	72.1
	CRF Map[213]	**86.9**	**90.2**	79.7	71.2	70.9	68.2	40.2	72.5
	RVSM	86.6	89.7	79.4	**73.9**	72.8	**70.0**	**42.3**	**73.3**

Table 3.3: KITTI Sequence 15 Compression Test for the frame 0-30. The mIoU is evaluated on the 15 ground truth images. In the compression experiment, query points are the centroids of the occupancy grid. Priors are discarded after the relevance vectors are obtained; thus, RVSM does not have access to prior locations but only to the positions of centroids in the occupancy grid.

Method	mIoU	No. of occupied voxels	No. of points/ voxels with labels	(Estimated) Size of semantic voxel maps
Prior Map	73.6	2499662	2499662	109.985 MB
RVSM	72.2	2499662	473	40.002 MB

from the camera for all approaches. For the CRF map, we use the code provided by the authors. The quantitative results are given in Table 3.2. *Average* represents the mean over all the classes. Although the RVSM is compressed, it still outperforms other maps on average recall and mean IoU while having comparable precision.

Comparison on multi-resolution performance: Using the same setup as above, we change different grid sizes, ranging from $0.1m$ to $1m$, when fusing semantic query results for the occupancy map for the prior, the baseline, and RVSM, as shown in Figure 9. The RVSM compression is done once, and the relevance vectors are held fixed for different resolutions, while in the baseline's result, one CRF post-processing is applied for each different resolution. Except for a grid size of $0.9m$, RVSM demonstrates better overall IoU on all other grid resolutions.

Compression evaluation: Table 3.3 shows the compression and query results for the first 31 frames in KITTI Sequence 15. Every third frame is used as a prior; we transform them into the global frame for RVSM compression. After obtaining 473 relevance vectors for all classes, we query on the voxel centroids of the corresponding occupancy map of resolution $0.1m$ for every frame and store the query results in the scrolling semantic occupancy map. In the table, the mIoU is evaluated on the 15 images that have ground truth of the total 31 prior images. The number of occupied voxels is computed from the scrolling occupancy map, counting how many voxels have adequate occupancy log-likelihood. If the map fusion works directly on the priors, such as the baseline, we would need to use the same number of voxels to store the distribution of all the classes in the map, one for each voxel. In contrast, RVSM only stores the relevance vectors, resulting in a more compact way of storing semantic information.

The estimated size of the semantic map is computed as follows:

- N_o: Number of occupied grid cells. Each grid cell has 4 floating point numbers, $(x, y, z, \text{occupancy})$
- n_f: Size of a floating point (in Bytes)
- N_s: Number of grid cells that have semantic label distribution
- n_c: Number of semantic classes
- N_{rv}: Number of relevance vectors.

With these definitions, the size of the semantic maps fused directly from the prior can be computed by $N_o \times (n_f \times 4) + N_s \times (n_f \times n_c)$. Meanwhile, the size of the semantic maps generated by fusing RVSM queries is $N_o \times (n_f \times 4) + N_{rv} \times (n_f \times 4)$. The second term is changed because RVSM computes the class of a voxel by (3.1), while each relevance vector v has 4 floating base numbers: (x_v, y_v, z_v) and weight w_v.

On this test sequence, RVSM has over 60% less memory consumption than directly fusing the map at the cost of 1.4% mIoU drop. This is an interesting trade-off when building autonomous systems that are much smaller than a passenger car, such as a bipedal robot.

3.5.3 NCLT Dataset

To evaluate the RVSM's performance on a sparse lidar point cloud, we use the NCLT dataset [22]. It contains 27 sequences collected on the University of Michigan's North Campus, with synchronized Velodyne HDL-32E 3D Lidar, an omnidirectional Ladybug3 camera, and proprioceptive odometry sensors [22]. To obtain ground truth semantic labels, we deployed LabelMe [149] annotation tools on Amazon Mechanical Turk and labeled 1194 training and 457 validation segmented images from NCLT's Ladybug3 camera. Each image frame is synchronized with one laser scan, covering the full $360°$ field of view. The projection of lidar points onto the segmented images yields their labels. The labeled classes are: *background, water, road, sidewalk, terrain, building, vegetation, car, person, bike, pole, stair, sign, and sky*. We perform a qualitative experiment on a subset of the 2012-04-29 sequence, as shown in Figure 10.

Table 3.4: NCLT Experiment RVM Model Selection: We run sequential RVM inference for 1000 iterations per class on 0.05 of the training data. The current kernel and hyperparameters are chosen using sample data and training a Gaussian process [143].

Kernel	Length-Scales (x, y, z)	Signal Variance	Iterations	Downsample Rate
Matern ($\nu = 5$) [143]	$(3.8096, 3.8096, 1.9090)$ m	0.8	1000	0.05

The data processing pipeline is similar to the KITTI experiment except for the way of obtaining priors and the tool used for map fusion. The prior distributions of the LiDAR point cloud are obtained from a 2D deep neural network generating predictions from the omnidirectional camera. Starting from the pretrained weight from Cityscapes [40], we fine-tuned the 2D segmentation network DeeplabV3+ [25] on the training set and reached 0.553 mIoU on the 2D validation set. Given NCLT's camera-lidar transformation, for each lidar scan, we project it onto the five image planes of the Ladybug camera so as to obtain the lidar point's prior distribution. Transformed by NCLT's ground truth poses, these priors are compressed into RVSM and then fused into a semantic octomap, which is the regular occupancy octomap [84] equipped with Bayesian label fusion. RVM model selection for the NCLT experiment is given in Table 3.4.

Figure 10 shows a top view of the route from Google Earth and the corresponding RVSM predictions. Here we use a grid size of $0.2m$ for the semantic Octomap. The result shows the ability of RVSM to recognize and reconstruct most of the categories in a large-scale mapping application. It also demonstrates the feasibility of RVSM on sparse LiDAR data.

3.6 Conclusion and Future Work

We proposed a compressed 3D semantic map representation that exploits sparse Bayesian modeling and a pipeline to build 3D semantic occupancy grid maps. We compared the prediction performance with other 3D semantic mapping systems and showed the continuity and compression properties of the proposed map representation. The experiments showed the proposed RVSM has comparable performance with the state-of-the-art techniques while requiring a lower memory footprint. The proposed semantic mapping system can be used in robotic applications where the semantic prediction needs to be made efficiently with limited onboard resources.

The optimization of the hyperparameters and model selection (selection of the best kernel) are interesting future directions. In the current implementation, the hyperparameters (parameters of the kernel) are optimized using a sample dataset and training a Gaussian process. While this method produces the presented results, the joint optimization of the hyperparameters during RVM training

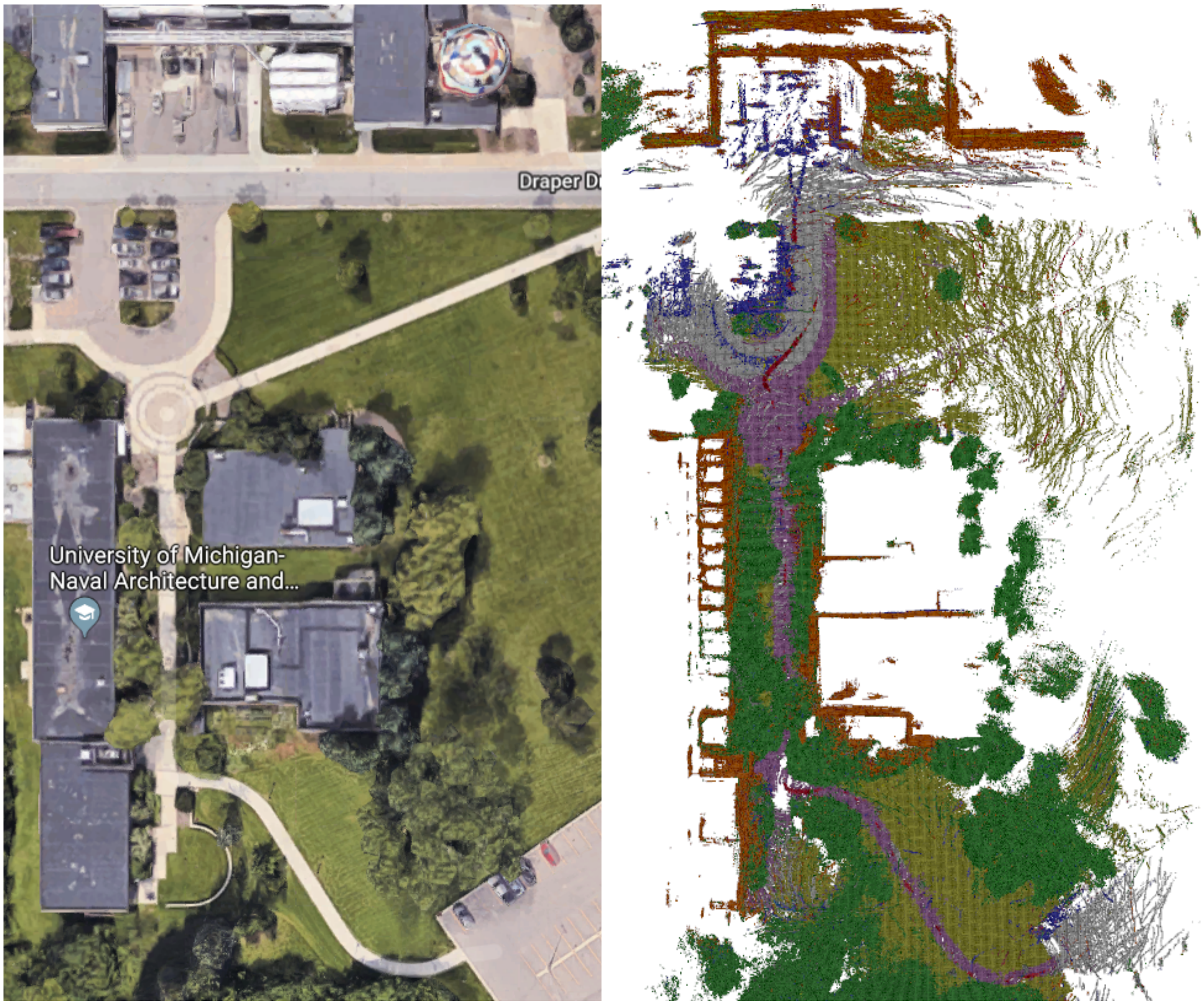


Figure 10: RVSM semantic map fusion on NCLT subsequence 2012-04-29. Left: The top-down view of the path from Google Earth. Right: Semantic Octomap fusion result after querying from RVSM.

is more convenient.

CHAPTER 4

SemanticCVO: A New Framework for Registration of Semantic Point Clouds from Stereo and RGB-D Cameras

4.1 Overview

This chapter reports on a novel nonparametric rigid point cloud registration framework, Semantic Continuous Visual Odometry (CVO), that jointly integrates geometric and semantic measurements such as color or semantic labels into the alignment process and does not require explicit data association. The point clouds are represented as nonparametric functions in a reproducible kernel Hilbert space. The alignment problem is formulated as maximizing the inner product between two functions, essentially a sum of weighted kernels, each of which exploits the local geometric and semantic features. As a result of the continuous models, analytical gradients can be computed, and a local solution can be obtained by optimization over the rigid body transformation group. Besides, we present a new point cloud alignment metric that is intrinsic to the proposed framework and takes into account geometric and semantic information. The evaluations using publicly available stereo and RGB-D datasets show that the proposed method outperforms state-of-the-art outdoor and indoor frame-to-frame registration methods. An open-source GPU implementation is also provided.

4.2 Introduction

Point cloud registration estimates the relative transformation between two noisy point clouds [9, 28, 159, 137, 29]. Point clouds obtained by RGB-D cameras, stereo cameras, and LIDARs contain rich color and intensity measurements besides geometric information. The extra non-geometric information can improve the registration performance [162, 132, 133]. Deep learning can provide semantic attributes of the scene as measurements [107, 24, 231]. As illustrated in Figure 11, this work focuses on constructing a novel integrated framework to jointly process raw geometric and non-geometric information for point cloud registration.

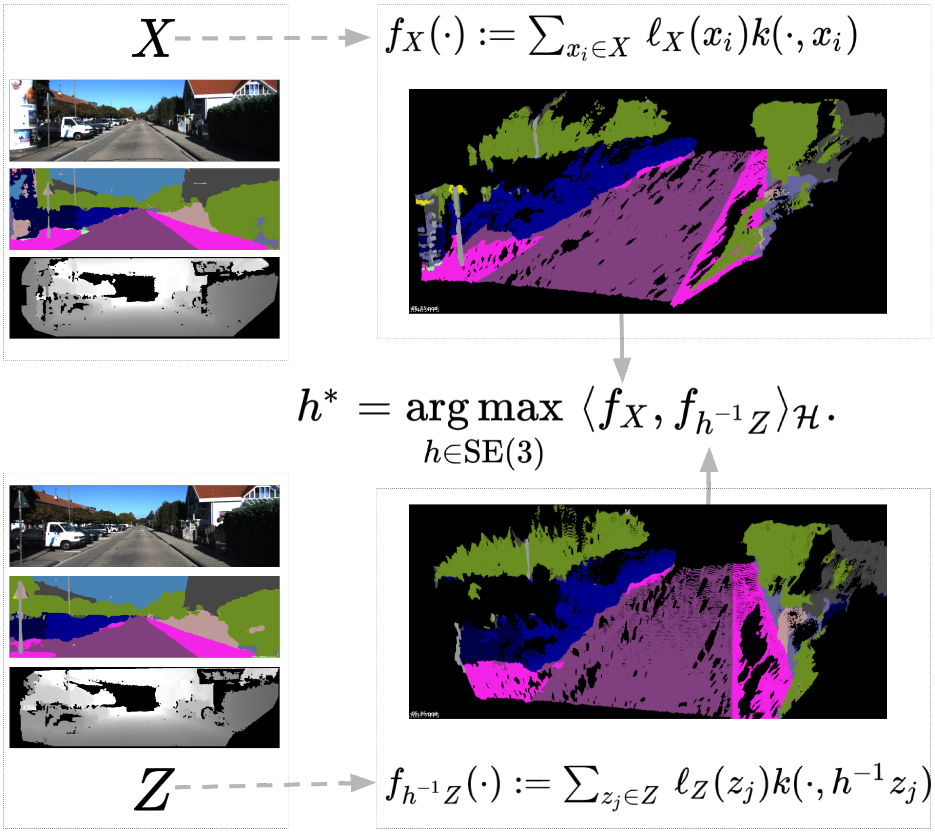

X
$f_X(\cdot) := \sum_{x_i \in X} \ell_X(x_i) k(\cdot, x_i)$
$h^* = \arg\max_{h \in \mathrm{SE}(3)} \langle f_X, f_{h^{-1}Z} \rangle_{\mathcal{H}}.$
Z
$f_{h^{-1}Z}(\cdot) := \sum_{z_j \in Z} \ell_Z(z_j) k(\cdot, h^{-1} z_j)$

Figure 11: Point clouds X and Z are represented by two continuous functions f_X, f_Z in a reproducing kernel Hilbert Space. Each point x_i has its semantic labels, $\ell_X(x_i)$, encoded in the corresponding function representation via a tensor product representation. The registration is formulated as maximizing the inner product between two point cloud functions.

Real-world applications such as 3D reconstruction [202] include noisy measurements, symmetries or dynamic objects, occlusion, and blurry observations. Examples are shown in Figure 12. These cases make the data association process challenging. Existing Iterative Closest Point (ICP)-based work [9, 28, 159] approach this problem by adding appearance/semantic features [131, 162, 132], adding local or deep geometric features [159, 32], and introducing weighted many-to-many correspondences [74, 76].

Gaussian Mixture Model (GMM) based registrations [11, 111, 91] model data correspondences and point clouds as probabilistic densities. Instead of point pairs, GMM-based methods work on point clusters, and the associations are a part of the model parameters. The relative rigid body transformation is then estimated by fitting the second point cloud measurements into the first point cloud's distributions [111, 33, 83, 54, 55, 56], or by minimizing a distance measure between the two distribution and inferring the weight parameters [185, 191, 12].

This work presents a nonparametric registration framework that jointly integrates geometric and semantic measurements and does not require explicit data association. Unlike existing methods that rely on geometric residuals with regularizers to include appearance information [131, 133], the proposed framework formulates the problem using a single objective function and is solved by the gradient ascent on Riemannian manifolds, similar to the work of [71, 34]. In particular, this work has the following contributions.

1. A novel framework for semantic point cloud registration that generalizes geometric, color, and semantic-assisted methods to a nonparametric continuous model via a hierarchical distributed representation of features.

2. A new point cloud alignment indicator that is intrinsic to the proposed framework and takes into account geometric and semantic information.

3. A global rotation initialization technique by evaluating the alignment measure under a finite set of discretized $\mathrm{SO}(3)$ elements.

4. An open source GPU implementation available at [222]:
https://github.com/UMich-CURLY/unified_cvo

5. Extensive evaluations using publicly available outdoor stereo and indoor RGB-D datasets.

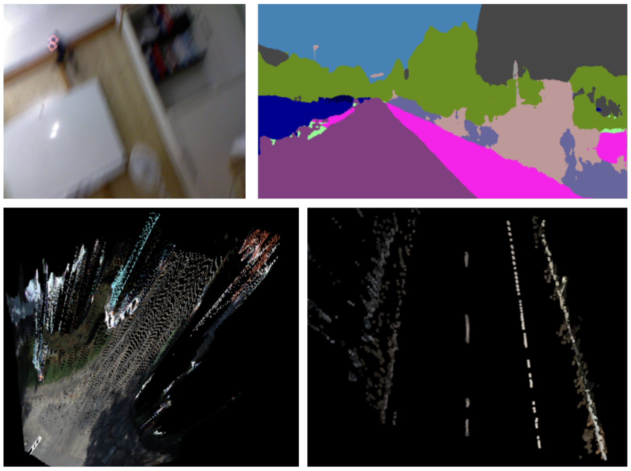


Figure 12: Challenging scenes for stereo and RGB-D point cloud registration, including blurry image sources (from TUM RGB-D [167]), noisy semantic sources (from KITTI [69] and Nvidia [231]), noisy depth estimations (from KITTI), and highly repetitive patterns (from KITTI).

4.3 Problem Setup

Consider two (finite) collections of points, $X = \{x_i\}$, $Z = \{z_j\} \subset \mathbb{R}^3$. We want to determine which element $h \in \mathrm{SE}(3)$, aligns the two point clouds X and $hZ = \{hz_j\}$ the "best." To assist with this, we will assume that each point contains the information described by a point in an inner product space, $(\mathcal{I}, \langle\cdot,\cdot\rangle_{\mathcal{I}})$. To this end, we will introduce two labeling functions, $\ell_X : X \to \mathcal{I}$ and $\ell_Z : Z \to \mathcal{I}$.

To measure their alignment, we turn the clouds, X and Z, into functions $f_X, f_Z : \mathbb{R}^3 \to \mathcal{I}$ that live in some reproducing kernel Hilbert space, $(\mathcal{H}, \langle\cdot,\cdot\rangle_{\mathcal{H}})$. The action, $\mathrm{SE}(3) \curvearrowright \mathbb{R}^3$ induces an action $\mathrm{SE}(3) \curvearrowright \mathcal{H}$ by $h.f(x) := f(h^{-1}x)$. Inspired by this observation, we will set $h.f_Z := f_{h^{-1}Z}$.

Problem 1. The problem of aligning the point clouds can now be rephrased as maximizing the scalar products of f_X and $h.f_Z$, i.e., we want to solve

$$\operatorname*{arg\,max}_{h\in\mathrm{SE}(3)} F(h), \quad F(h) := \langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}}. \tag{4.1}$$

4.3.1 Constructing The Functions

We follow the same steps in [71] with an additional step in which we use the kernel trick to kernelize the information inner product. For the kernel of our RKHS, $\mathcal{H}$, we first choose the squared exponential kernel $k : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$:

$$k(x,z) = \sigma^2 \exp\left(\frac{-\|x-z\|_3^2}{2\ell^2}\right), \tag{4.2}$$

for some fixed real parameters (hyperparameters) σ and ℓ (the *lengthscale*), and $\|\cdot\|_3$ is the standard Euclidean norm on $\mathbb{R}^3$. This allows us to turn the point clouds into functions via

$$\begin{aligned} f_X(\cdot) &:= \sum_{x_i\in X} \ell_X(x_i)k(\cdot,x_i), \\ f_{h^{-1}Z}(\cdot) &:= \sum_{z_j\in Z} \ell_Z(z_j)k(\cdot,h^{-1}z_j). \end{aligned} \tag{4.3}$$

Here, $\ell_X(x_i)$ encodes the appearance information, for example, LIDAR intensity and image pixel color. $k(\cdot,x_i)$ encodes the geometric information. We can now obtain the inner product of f_X and f_Z as

$$\langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}} := \sum_{x_i\in X, z_j\in Z} \langle \ell_X(x_i), \ell_Z(z_j)\rangle_{\mathcal{I}} \cdot k(x_i, h^{-1}z_j) \tag{4.4}$$

We use the kernel trick in machine learning[13, 143, 122] to substitute the inner products in (4.4) with the appearance kernel. After applying the kernel trick to (4.4), we get

$$\langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}} = \sum_{x_i \in X, z_j \in Z} k_c(\ell_X(x_i), \ell_Z(z_j)) \cdot k(x_i, h^{-1}z_j), \tag{4.5}$$

We choose k_c as the squared exponential kernel with real hyperparameters σ_c and ℓ_c that are set independently.

4.3.2 Feature Embedding via Tensor Product Representation

We now extend the feature space to a hierarchical distributed representation. Let $(V_1, V_2, \dots)$ be different inner product spaces describing different types of non-geometric features of a point, such as color, intensity, and semantics. Their tensor product, $V_1 \otimes V_2 \otimes \dots$, is also an inner product space. For any $x \in X, z \in Z$ with features $\ell_X(x) = (u_1, u_2, \dots)$ and $\ell_Z(z) = (v_1, v_2, \dots)$, with $u_1, v_1 \in V_1$, $u_2, v_2 \in V_2, \dots$, we have

$$\begin{aligned}\langle \ell_X(x), \ell_Z(z)\rangle_{\mathcal{I}} &= \langle u_1 \otimes u_2 \otimes \dots, v_1 \otimes v_2 \otimes \dots\rangle \\ &= \langle u_1, v_1\rangle \cdot \langle u_2, v_2\rangle \cdot \dots . \end{aligned} \tag{4.6}$$

By substituting (4.6) into (4.4), we obtain

$$\langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}} = \sum_{\substack{x_i \in X \\ z_j \in Z}} \langle u_{1i}, v_{1j}\rangle \cdot \langle u_{2i}, v_{2j}\rangle \dots k(x_i, h^{-1}z_j)$$

After applying the kernel trick, we arrive at

$$\begin{aligned}\langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}} &= \sum_{x_i \in X, z_j \in Z} k(x_i, h^{-1}z_j) \cdot \prod_k k_{V_k}(u_{ki}, v_{kj}) \\ &:= \sum_{x_i \in X, z_j \in Z} k(x_i, h^{-1}z_j) \cdot c_{ij}. \end{aligned} \tag{4.7}$$

Equation (4.7) describes the full geometric and non-geometric relationship between the two point clouds. Each c_{ij} does not depend on the relative transformation; thus, it will be a constant when computing the gradient and the step size. In our implementation, the double sum in (4.7) is sparse because a point $x_i \in X$ is far away from the majority of the points $z_j \in Z$, either in the spatial (geometry) space or one of the feature (semantic) spaces.

This formulation can be further simplified to a purely geometric model if we let the label functions $\ell_X(x_i) = \ell_Z(z_j) = 1$. Then (4.7) becomes

$$\langle f_X, f_{h^{-1}Z} \rangle_{\mathcal{H}} = \sum_{x_i \in X, z_j \in Z} k(x_i, h^{-1} z_j). \tag{4.8}$$

Through (4.8), the proposed method can register point clouds that do not have appearance measurements. It is worth noting that, when choosing the squared exponential kernel, (4.8) has the same formulation as Kernel Correlation [185].

4.3.3 An Indicator of Alignment

We want an indicator that represents the alignment of two point clouds, X and Z. An intrinsic metric available in our framework is the angle, θ, between two functions. This indicator can be computed to track the optimization progress. The cosine of the angle is defined as

$$\cos(\theta) = \frac{\langle f_X, f_Z \rangle_{\mathcal{H}}}{\|f_X\| \cdot \|f_Z\|}. \tag{4.9}$$

However, calculating $\|f_X\|$ and $\|f_Z\|$ is time-consuming as it requires evaluating the double sum for each of the two point clouds. To approximate (4.9), we use the following result.

Remark 1. Suppose $k(x_i, x_j) = \delta_{ij}$ and $c_{ii} = 1$, where δ_{ij} is the Kronecker delta, then $\|f_X\| = \sqrt{|X|}$.

Corollary 1. Using the previous assumption, we define the following alignment indicator.

$$i_\theta := \frac{1}{\sqrt{|X| \cdot |Z|}} \sum_{x_i \in X, z_j \in Z} c_{ij} \cdot k(x_i, z_j). \tag{4.10}$$

The behavior of the alignment indicator with respect to the rotation and translation errors is shown in Figure 13. We manually rotate and translate the same point cloud and then calculate the indicator with the original point cloud. A larger transformation results in a smaller indicator value. Furthermore, the maximum indicator value occurs when the transformation error is zero.

Remark 2. OverlapNet [26] uses a neural network to predict a similar metric and detect loop closures. The cosine of the angle in (4.9) or the indicator in (4.10) provide such a metric for *self-supervised* learning while taking into account the semantic information. Given the promising results of [26], combining our metric with deep learning is an interesting future research direction.

4.3.4 Optimization

We perform gradient ascent over $SE(3)$ on the inner product in (4.7). The flow and step size expressions are in the same forms as [71]. During the iterative optimization, the alignment indicator in 4.3.3 can guide the lengthscale update. When the lengthscale is large, each point is associated with farther points, which provides the point cloud function with a global perspective. When the lengthscale is small, each point is only connected to its closest neighbors, resulting in local attention for the registration. For a single registration process, we use larger lengthscales at early iterations. Every time the alignment indicator value at the current lengthscale stabilizes, we decay the lengthscale by a fixed percent.

4.4 Considerations for Boosting the Performance

The hyperparameters to be tuned include the lengthscale of the geometric and the appearance (color and semantic) kernels. We use the same hyperparameters across different sequences within a dataset. At the first frame of an entire data sequence, we initialize the transformation to be identity. As the initial motions of the robot are unknown, we use a large lengthscale only for this single frame (0.95 in all the KITTI Stereo geometric sequences) at the cost of more iterations. In subsequent frames, we use the previously estimated transformation as the initial value, accompanied by a smaller starting lengthscale (0.1 in all KITTI).

To address the costly double-sum computation, we downsample the raw inputs. We adopt the FAST feature detector [147] implemented in OpenCV [17]. We automatically control FAST's threshold of the pixel intensity difference and disable the non-max suppression so that the number of selected pixels with non-empty depth values is between 3k and 15k.

4.5 Global Registration with Unknown Initialization

We perform pose optimization based on gradient ascent, which assumes a good initial guess that is not far away from the ground truth. However, there are no immediate initial guesses in real applications such as loop closure registrations and robot relocalizations. Furthermore, first-order-based optimization techniques easily fall into local minima due to the non-convex objective functions in Eq. (4.7) as well as the non-convex structure of the $\mathrm{SO}(3)$ manifold.

To mitigate the issue of local minima, we leverage the property that the objective function in Eq. (4.7) is a continuous function with respect to the poses. Subsequently, we discretize the $\mathrm{SO}(3)$ group into a finite number of uniformly-distributed rotations. Then, we measure the quality of each initial pose guess using the alignment measure in Eq. (4.9). The rotation demonstrating the

Table 4.1: Results of the proposed frame-to-frame method using the KITTI [69] stereo odometry benchmark as evaluated on the average drift in translation, as a percentage (%), and rotation, in degrees per meter. The drifts are calculated for all possible subsequences of $100, 200...., 800$ meters.

		00	01	02	03	04	05	06	07	08	09	10	**Avg**	**Std**
GeometricCVO	t (%)	4.06	7.04	5.86	**3.84**	5.08	3.42	**2.99**	5.23	4.40	4.67	**3.42**	4.55	1.20
	r (m)	0.0173	0.0285	0.0220	0.0199	0.0358	0.0206	0.0151	0.0444	0.0188	0.0185	0.0181	0.0236	0.00907
GICP [159]	t (%)	8.66	26.19	7.92	7.64	7.40	6.06	16.40	8.45	14.69	7.35	12.73	11.23	5.99
	r (m)	0.0361	0.0467	0.0302	0.0460	0.0548	0.0336	0.0616	0.0657	0.0453	0.0248	0.0525	0.0452	0.0130
3D-NDT [111]	t (%)	7.53	16.41	6.11	5.13	4.63	6.76	11.68	11.16	7.67	5.50	10.96	8.50	3.63
	r (m)	0.0388	0.0272	0.0261	0.0432	0.0302	0.0346	0.0472	0.0791	0.0387	0.0237	0.0467	0.0396	0.0155
ColorCVO	t (%)	**3.19**	4.42	5.00	3.94	3.86	**2.94**	3.18	**2.32**	**3.65**	4.39	3.64	3.69	0.76
	r (m)	**0.0125**	0.0158	0.0167	0.0182	0.0230	0.0152	**0.0103**	0.0176	0.0147	0.0151	0.0154	0.0159	0.00323
MC-ICP [162]	t (%)	7.77	55.26	11.33	15.45	9.65	5.51	9.65	13.62	6.54	8.16	12.16	14.10	13.98
	r (m)	0.0387	0.0598	0.0357	0.0749	0.0585	0.0335	0.0335	0.0927	0.0314	0.0277	0.0504	0.0488	0.0208
SemanticCVO	t (%)	3.22	**3.97**	**4.96**	3.94	**3.84**	2.95	3.28	2.35	**3.65**	**4.32**	3.59	**3.64**	**0.70**
(with color)	r (m)	0.0126	**0.0132**	**0.0166**	**0.0179**	**0.0227**	**0.0150**	0.0105	**0.0172**	**0.0146**	**0.0148**	**0.0151**	**0.0155**	**0.00321**

maximum alignment value is designated as the initial rotation.

4.6 Experimental Results

We first present a controlled simulated experiment of global pose regression with the Stanford Bunny Dataset [186]. Second, we present the local frame-to-frame registration experiments on both outdoor and indoor real datasets: KITTI stereo odometry [69] and TUM RGB-D dataset [167].

4.6.1 Experimental Setup

The simulated experiment performs large-angle two-frame registration of the bunny scan, initialized from identity. The baselines include two global registration techniques, RANSAC [62] and FGR [226]. All the three methods use the point-wise FPFH [150] features.

The real data experiments are performed in a frame-to-frame manner without skipping images. The first frame's transformation is initialized with identity, and all later frames start with the previous frames' results. Hyperparameters for the proposed method on KITTI stereo and TUM RGB-D are available at [222]. The same values were used within one dataset.

On KITTI, our baselines are GICP [159], Multichannel-ICP [162], and 3D-NDT [111]. GICP and NDT are compared with our geometric method (*Geometric CVO*). Multichannel-ICP competes with our color-assisted method (*Color CVO*). GICP and 3D-NDT implementation are from PCL [151]. The Multichannel-ICP implementation is from [133]. Both the baselines and the proposed methods remove the first 100 rows of image pixels, mainly sky pixels, and points that are more than 55 meters away. All the baselines use full point clouds without downsampling. The discussions of more candidate baselines and point selectors are in Sec. 4.7

On TUM RGB-D, we use the same baselines for geometric registration as KITTI. We compare Color CVO with Dense Visual Odometry (DVO) [96] and Color ICP [131]. We reproduced DVO results with the code from [136] because the original DVO source code requires an outdated ROS dependency [95]. The Color ICP implementation is taken from Open3D [227]. The baselines use full point clouds.

4.6.2 Simulated Example: Global Rotation Initialization

We use the Standford Bunny point cloud scan [186] to test the global initialization under different rotation configurations. The two point clouds are initialized as follows. First, they are randomly rotated with two different angles, $90°$ and $180°$, along a random axis. Second, random translations with length 0.5 are further applied. Third, we perturb the point clouds with point-wise Gaussian mixture noises. It has five different outlier ratios: $0\%, 12.5\%, 25\%, 37.5\%,$ and 50%. If it is sampled as an inlier, then we add a Gaussian perturbation $\mathcal{N}(0, 0.01)$ along the normal direction of the point. If it is an outlier, we also add a uniform noise between $(-0.1, 0.1)$ along the point's normal direction. Last but not least, we randomly crop $0\%, 12.5\%, 25\%, 37.5\%,$ and 50% of the two point clouds so that they do not fully overlap.

Figure 15 shows the qualitative results of the proposed global rotation initialization versus the baselines, under 50% uniformly distributed outliers and 50% random cropping, when an unknown pose with 180^{circ} rotation is imposed. The initial data pair has fewer than 50% overlap. Under such perturbations, one-to-one data correspondence is challenging for classical methods. The proposed method can retrieve the correct transformation while the baselines cannot.

Figure 16 and Figure 17 show the quantitative results of the proposed global rotation initialization versus the baselines when unknown poses with 90^{circ} and 180^{circ} rotation are imposed, under a range of various outlier ratio and cropping ratio. The proposed method can retrieve the correct transformation compared to the baselines. Under such large angles, the baselines cannot correctly regress the correct transformation. In contrast, the proposed method has a relatively low error ($< 1e^{-2}$) when the cropping ratio is less than 37.5%. The errors increase significantly when the cropping ratio reaches 50% at both angles. The two figures show the proposed method's superior robustness under large angles and the existence of outliers.

Table 4.2: The RMSE of Relative Pose Error (RPE) for `fr1` sequences. The trans. columns show the RMSE of the translational drift in $\mathrm{m/\,sec}$ and the rot. columns show the RMSE of the rotational error in $\mathrm{deg/\,sec}$.

	Geometric CVO		GICP[159]		3D-NDT[111]		Color CVO		DVO[96]		Color ICP[131]	
Sequence	Trans.	Rot.	Trans.	Rot.	Trans.	Rot.	Trans.	Rot.	Trans.	Rot.	Trans.	Rot.
fr1/desk	0.0493	2.3377	0.2358	11.9360	0.2404	13.5183	0.0384	**2.1422**	0.0387	2.3589	0.0938	5.2660
fr1/desk2	0.0545	**2.7190**	0.3617	19.8483	0.1823	11.8914	**0.0515**	2.8967	0.0518	3.6529	0.2304	8.5799
fr1/room	0.0565	**2.2946**	0.3966	17.0337	0.1718	9.9076	**0.0501**	2.3366	0.0518	2.8686	0.1444	6.2150
fr1/360	**0.1001**	**2.8686**	0.5251	17.0537	0.2245	13.6262	0.1021	3.1086	0.1602	4.4407	0.2325	8.6135
fr1/teddy	0.0663	**2.4122**	0.4659	16.3678	0.2095	11.2214	0.0668	2.6016	0.0948	2.5495	0.1735	5.7976
fr1/floor	0.2267	2.7345	0.2008	6.5601	0.5560	35.9573	0.0697	2.3663	**0.0635**	**2.2805**	0.0668	3.3416
fr1/xyz	**0.0238**	**0.9748**	0.1093	7.8490	0.1102	5.5953	0.0270	1.1379	0.0327	1.8751	0.0632	4.5334
fr1/rpy	0.0413	3.1806	0.4802	19.4342	0.2329	16.8113	0.0501	3.6598	0.0336	**2.6701**	0.0930	5.8095
fr1/plant	0.0388	1.9027	0.8551	26.8711	0.1335	7.7507	0.0347	1.6451	**0.0272**	**1.5523**	0.1205	4.9295
Average	0.0730	**2.3805**	0.4034	15.8838	0.2290	14.0311	**0.0545**	2.4333	0.0623	2.6943	0.1353	5.8985

4.6.3 Outdoor Stereo Camera: KITTI Stereo Odometry

We select a subset of pixels from KITTI's stereo images via OpenCV's FAST [147] feature detector. The depth values of the selected pixels are generated with ELAS [70]. The semantic predictions of the images come from Nvidia's pre-trained neural network [231], which was trained on 200 labeled images. Examples of the point clouds are in Figure 12. Noise from the estimated depth, from the color sensor, and from the semantic predictions are visible.

The result of Geometric, Color, and Semantic CVO and other baselines are provided in Table 4.1. From sequence 00 to 10, our geometric method has a lower average translational error (4.55%) compared to the GICP (11.23%) and NDT (8.50%). Our color version has a lower average translational drift (3.69%) than Multichannel-ICP (14.10%). The error is further reduced if we add semantic information (3.64%). The addition of color and semantic information also yields a lower standard deviation. Meanwhile, the average rotational drift of the proposed methods is smaller. Specifically, on the highway sequence (01), where the point cloud pattern becomes repetitive and noisy, both NDT and GICP perform poorly (as shown in Figure 14). Figure 18 shows the average translational and rotational errors at different distances and speeds. The proposed methods offer a more consistent high accuracy across different speeds.

On our desktop computer, excluding the image I/O and point cloud generation operations, the current GPU implementation takes, on average, $1.4\,\mathrm{sec}$ per frame when registering less than 15k points after being downsampled with FAST point selector. GICP, NDT, and Multichannel-ICP use full point clouds (150k-350k points) and take $6.3\,\mathrm{sec}$, $6.6\,\mathrm{sec}$, and $57\,\mathrm{sec}$ per frame on CPU, respectively.

Table 4.3: The RMSE of Relative Pose Error (RPE) for the structure v.s texture sequence. The Trans. columns show the RMSE of the translational drift in $\mathrm{m}/\sec$ and the Rot. columns show the RMSE of the rotational error in $\deg/\sec$.

			Geometric CVO		GICP[159]		3D-NDT[111]		Color CVO		DVO[96]		Color ICP[131]	
structure-texture			Trans.	Rot.	Trans.	Rot.	Trans.	Rot.	Trans.	Rot.	Trans.	Rot.	Trans.	Rot.
×	✓	near	0.0267	0.8745	0.2602	7.5238	0.4586	13.4089	0.0250	**0.8201**	0.0563	1.7560	**0.0212**	0.9744
×	✓	far	**0.0498**	1.1602	0.3115	3.3421	0.2034	4.8534	0.0591	**1.1393**	0.1612	3.4135	0.0755	1.6356
✓	×	near	0.0338	2.4081	0.0628	2.0061	0.0993	5.5899	0.0505	3.5577	0.1906	10.6424	0.0255	**1.0317**
✓	×	far	**0.0376**	1.2435	0.1172	3.6457	0.0861	1.8595	0.0456	**1.2239**	0.1171	2.4044	0.0592	1.7822
✓	✓	near	0.0238	1.3058	0.1573	6.0924	0.1082	4.6971	0.0344	1.6899	**0.0175**	**0.9315**	0.0200	1.2008
✓	✓	far	0.0288	0.9314	0.1921	4.6908	0.0717	1.9343	0.0293	0.9516	**0.0171**	**0.5717**	0.0434	1.1375
×	×	near	0.3057	10.8878	0.3685	12.6208	0.5901	16.1501	0.2143	8.9564	0.3506	13.3127	**0.2064**	**7.7856**
×	×	far	**0.1287**	4.0173	0.2232	2.4611	0.3722	7.3946	0.1449	2.9821	0.1983	6.8419	0.2052	**2.0850**
Average			0.0794	2.8536	0.2116	5.2979	0.2487	6.9860	**0.0754**	2.6651	0.1386	4.9843	0.0820	**2.2041**

4.6.4 Indoor RGB-D Camera: TUM RGB-D Dataset

For TUM RGB-D, a semi-dense point cloud is generated from the depth images with FAST [147] feature selector. We evaluated our method on the `fr1` sequences, which are recorded in an office environment, and `fr3` sequences, which contain image sequences in structured/nostructured and texture/notextured environments. TABLE 4.2 shows the results of `fr1` sequences. Geometric CVO outperforms the baselines and achieves a similar performance to DVO. Moreover, with color information, the average error of CVO decreases.

The results of `fr3` sequences are shown in TABLE 4.3. CVO outperforms the baselines. The overall result of Color CVO is better than Geometric CVO. However, Geometric CVO has lower translation errors in some sequences. This might be caused by the motion blur in the image, where color information is noisy due to rapid camera motion.

4.7 Discussions and Limitations

Besides the reported baseline results, we also tried to run Semantic ICP [132] and Color ICP[131] with KITTI's full stereo point clouds. However, Semantic-ICP takes $4 - 8\,\mathrm{min}$ per frame on our machine, so completing all the 23190 frames is infeasible. The original Color ICP work was not tuned for stereo data and failed on KITTI sequences 00 and 08. We also tried to use the FAST point selector for all the baselines, but only GICP shows improvements, with 7.98% translation drift and $0.0362\,\mathrm{m}$ rotation drift, versus our geometric result being 4.55% and $0.0236\,\mathrm{m}$.

We noticed that the point selector has a significant influence on the performance of the proposed methods. DSO's semi-dense point selector in [59] was unable to complete some challenging sequences, such as KITTI sequence 01. We cannot use PCL's Voxel Filter[151] either because the original color and semantic information are lost during its downsampling. Only FAST[147] feature selector from OpenCV[17] works for all the tested datasets. A future direction is to find a more robust downsampling scheme for this framework.

Moreover, the performance of the proposed methods relies on the geometric lengthscale during the optimization. Adaptive CVO[104] addresses the lengthscale decay by regarding it as a part of the optimizing variable. Still, we need to choose an initial lengthscale manually. The lengthscale needs a global perspective for such abrupt changes for inputs with larger accelerations. In this case, another future direction is an algorithmic way of selecting the initial lengthscale or, more broadly, studying the hyperparameter learning problem.

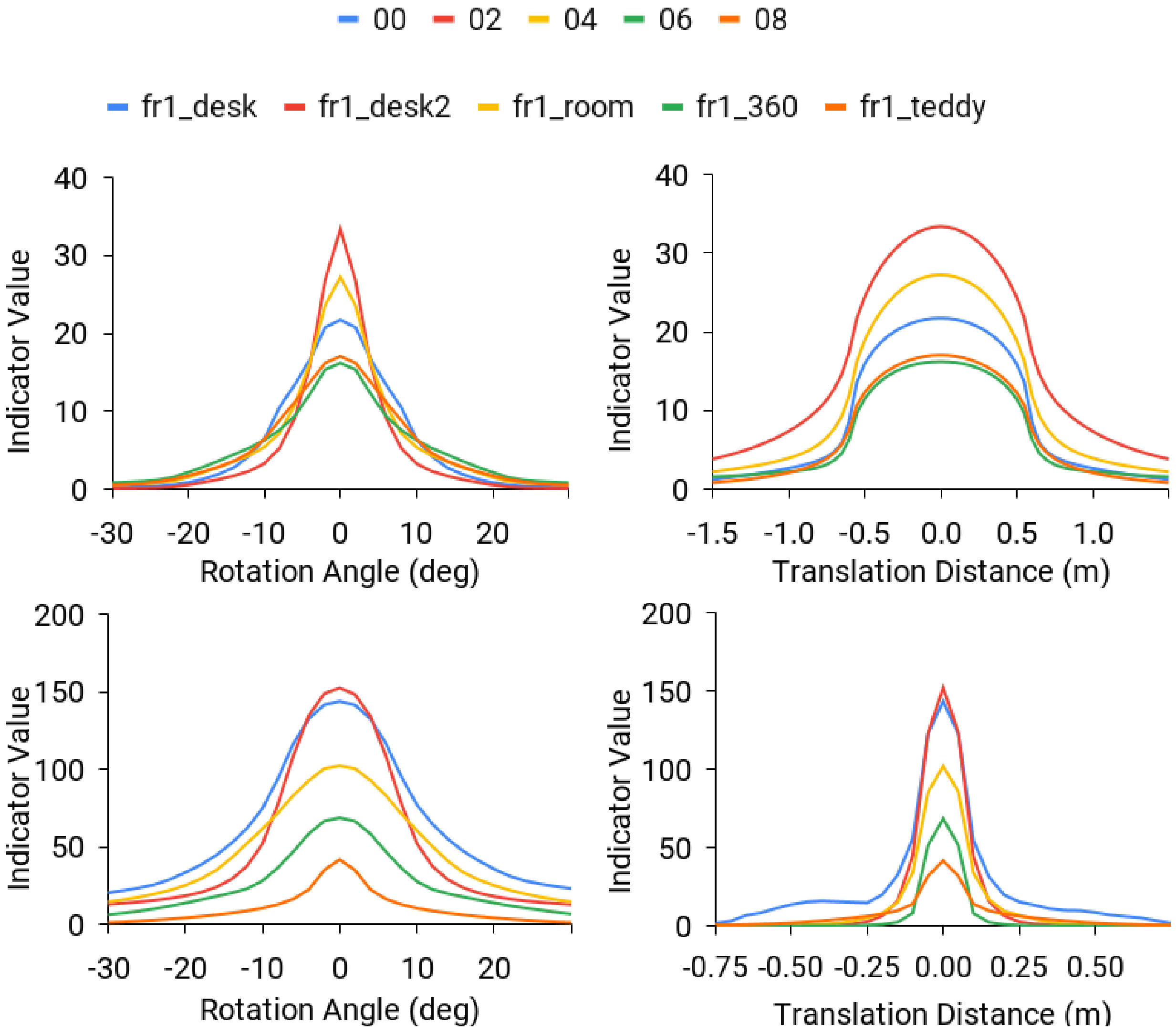


Figure 13: Indicator value with respect to rotation angle and translation distance for KITTI Stereo (left figures) and TUM RGB-D (right figures) sequences.

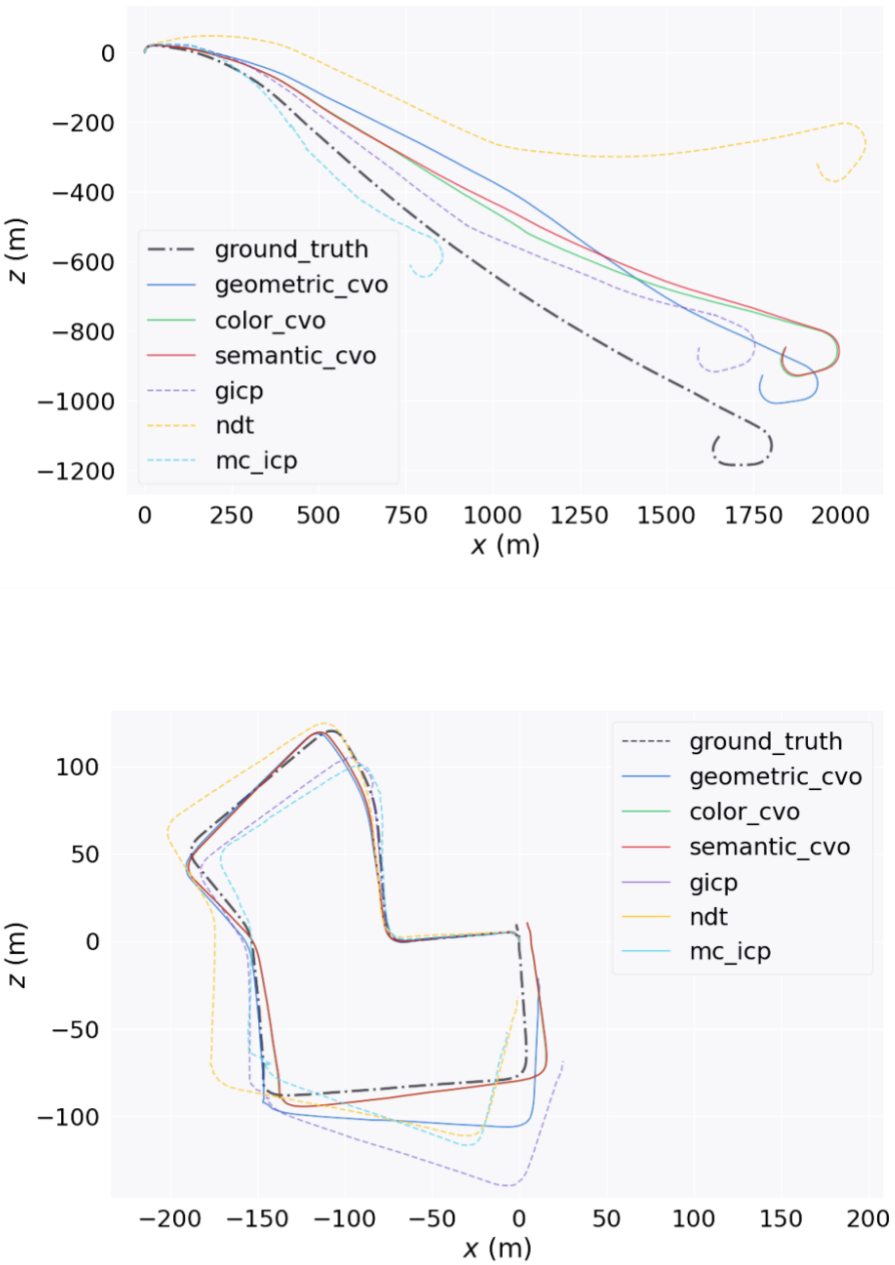
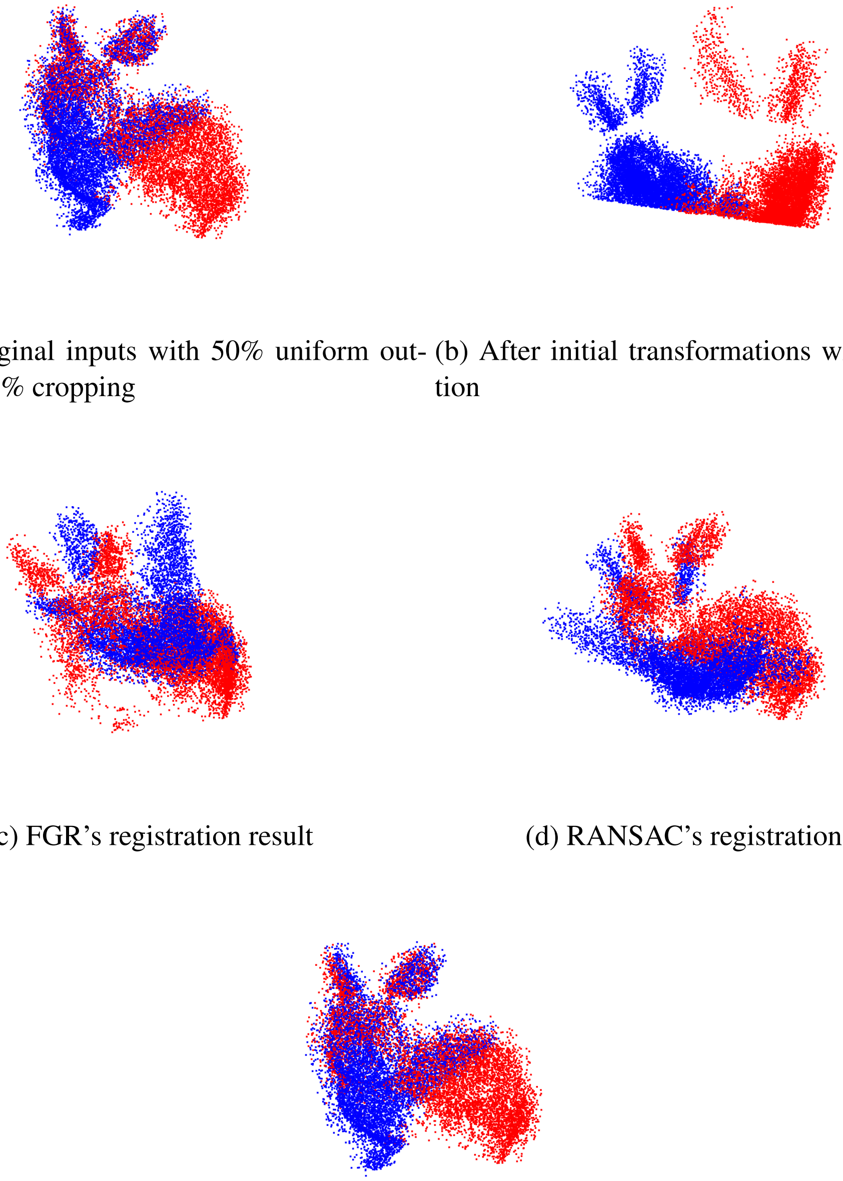


Figure 14: An illustration of the proposed registration method on KITTI stereo sequence 01 (top) and 07 (bottom) versus the baselines. The black dashed trajectory is the ground truth. The dot-dashed trajectories are the baselines. Plotted with EVO[78].



(a) The original inputs with 50% uniform outliers and 50% cropping

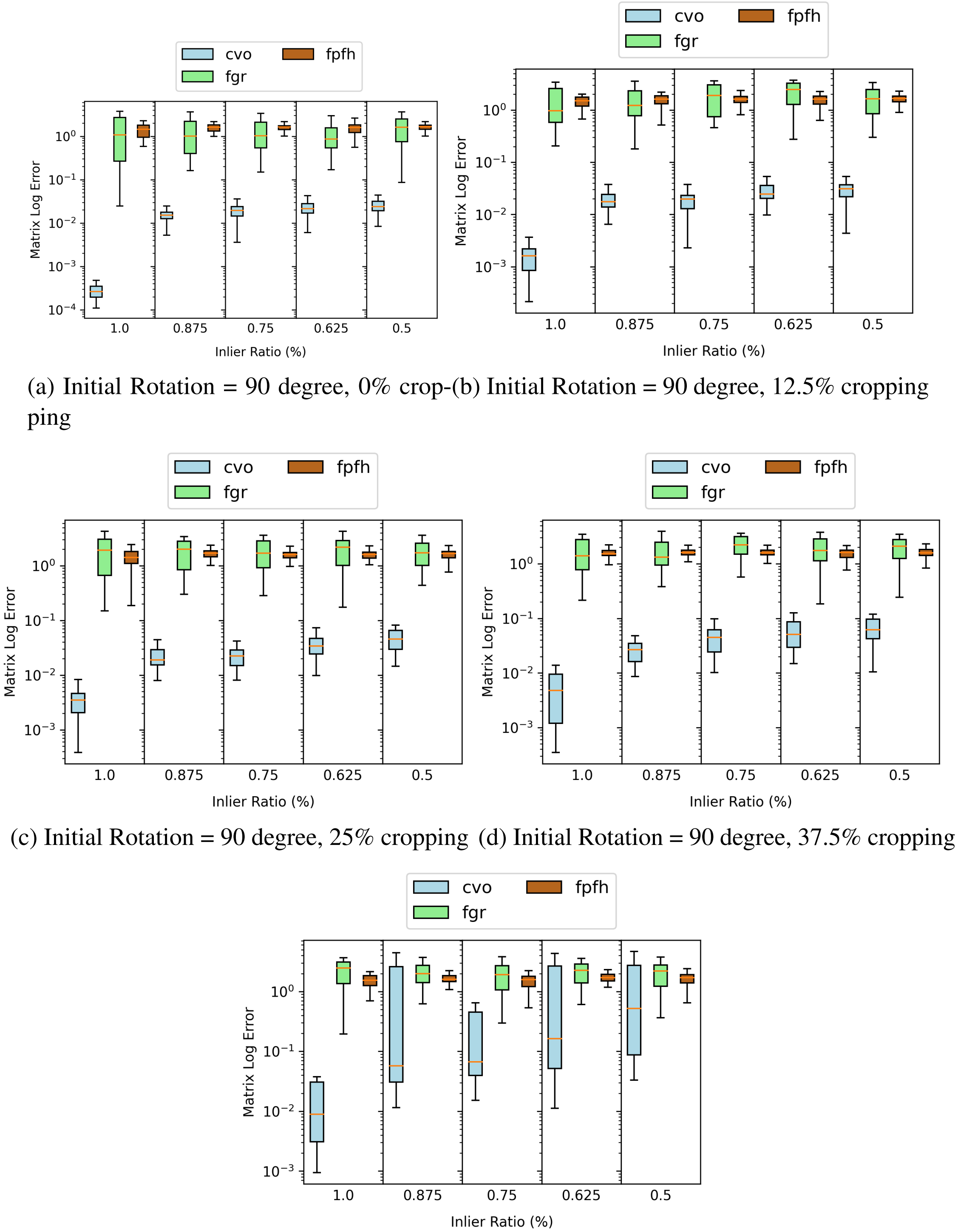
(b) After initial transformations with 180° rotation

(c) FGR's registration result

(d) RANSAC's registration result

(e) The proposed method's registration result with global rotational initialization

Figure 15: An example of a two-view point cloud registration test with FPFH invariant feature information on the Bunny [186] Dataset. (a) The two partially overlapped point clouds of the Bunny Dataset, each perturbed by 50% random outliers and 50% cropping. (b) The two Bunny point clouds after we apply initial rotations of 180 degrees around a random axis and a random translation of $0.5m$. (c) FGR's registration result. (d) RANSAC's registration result. (e) The proposed method's registration results using global rotational initialization.



(a) Initial Rotation = 90 degree, 0% cropping

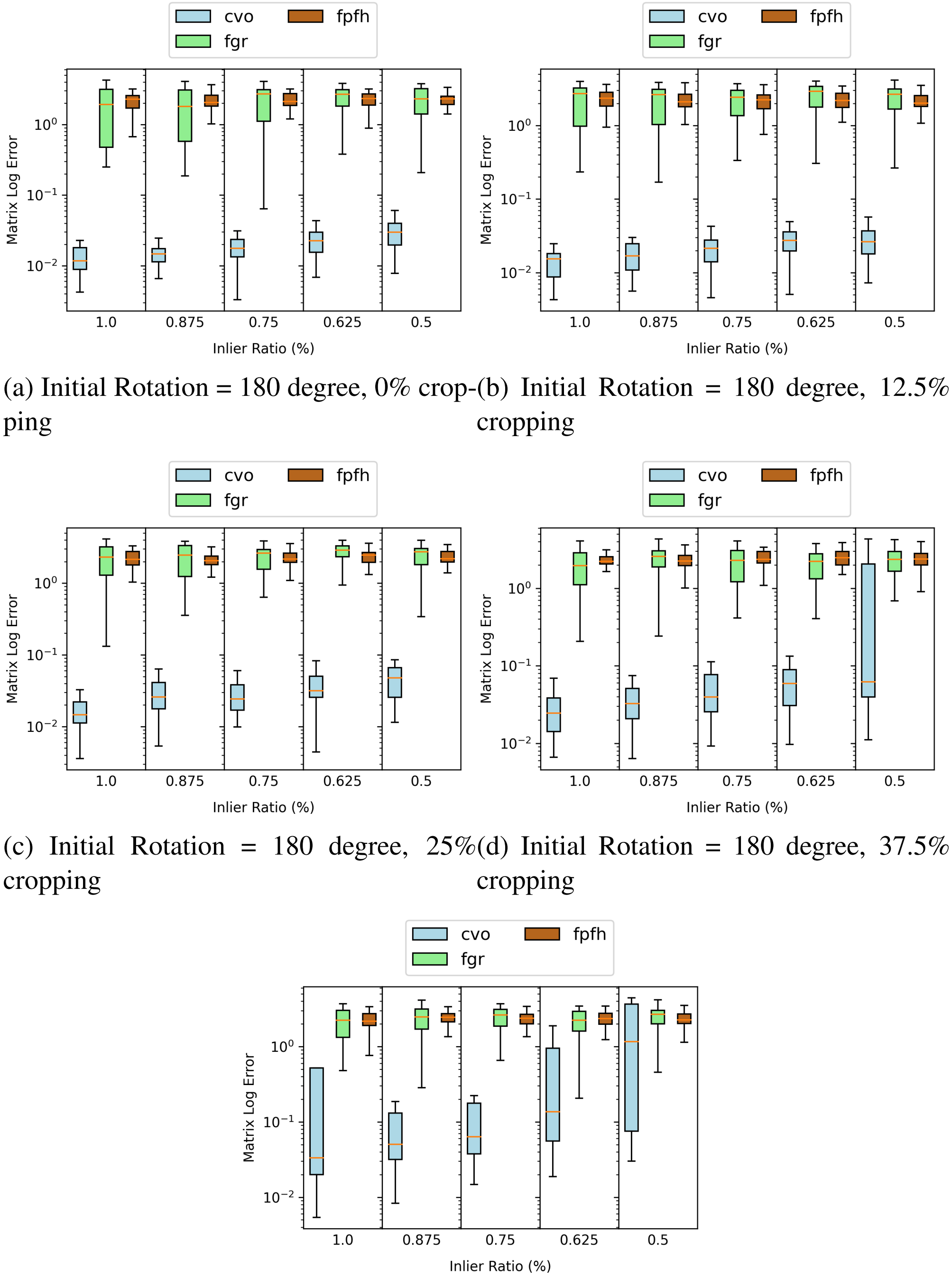
(b) Initial Rotation = 90 degree, 12.5% cropping

(c) Initial Rotation = 90 degree, 25% cropping

(d) Initial Rotation = 90 degree, 37.5% cropping

(e) Initial Rotation = 90 degree, 50% cropping

Figure 16: The benchmark results of the two registration tests on the Bunny Dataset [186]. Each box plot contains the resulting pose errors in the norm of matrix logarithm under different outlier ratios and cropping ratios at the same 90° initial rotation angle. (a) 0% cropping (b) 12.5% cropping (c) 25% cropping (d) 37.5% cropping (e) 50% cropping



(a) Initial Rotation = 180 degree, 0% cropping

(b) Initial Rotation = 180 degree, 12.5% cropping

(c) Initial Rotation = 180 degree, 25% cropping

(d) Initial Rotation = 180 degree, 37.5% cropping

(e) Initial Rotation = 180 degree, 50% cropping

Figure 17: The benchmark results of the two-view registration on the Bunny Dataset [186]. Each box plot contains the resulting pose errors in the norm of matrix logarithm under different outlier ratios and cropping ratios at the same 180° initial rotation angle. (a) 0% cropping (b) 12.5% cropping (c) 25% cropping (d) 37.5% cropping (e) 50% cropping

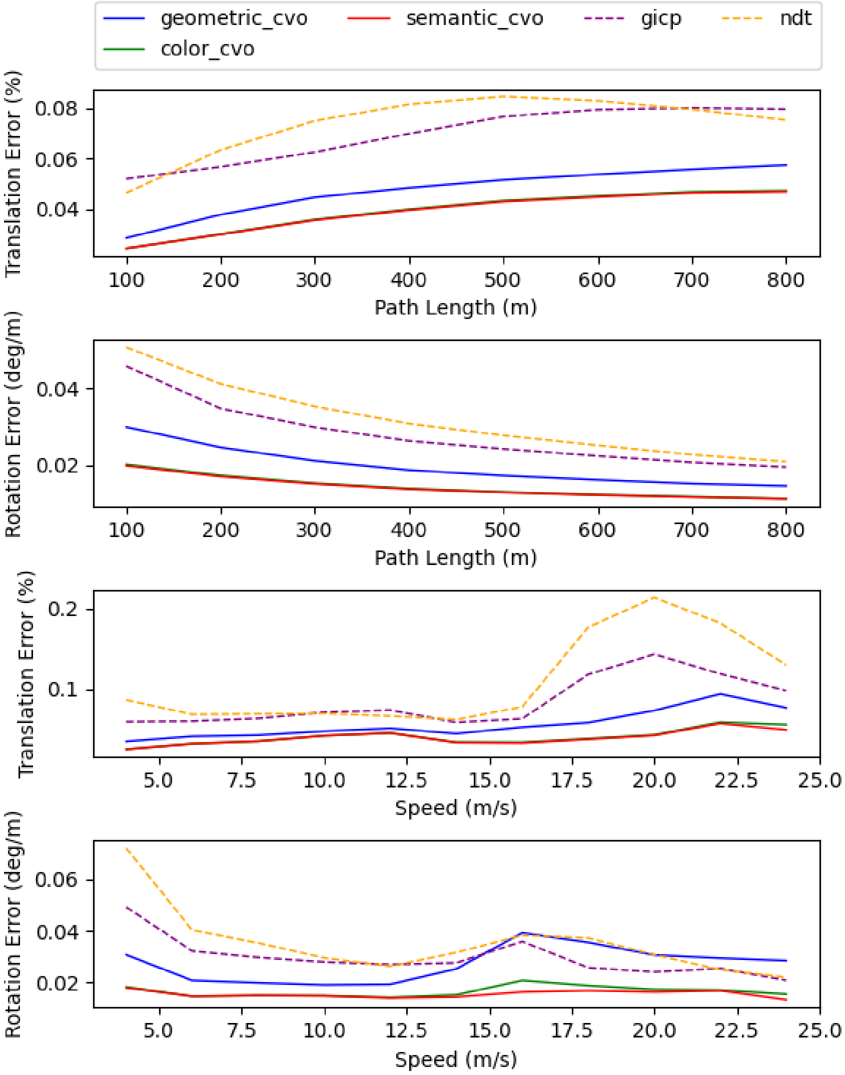


Figure 18: From top to down: the average translation errors and rotation errors on KITTI Stereo sequences 00 to 10 with respect to the distance segment and the moving speed, respectively.

CHAPTER 5

RKHS-BA: A Semantic Correspondence-Free Multi-View Registration Framework

5.1 Overview

This work reports a novel Bundle Adjustment (BA) formulation using a Reproducing Kernel Hilbert Space (RKHS) representation called RKHS-BA. The proposed formulation is correspondence-free, enables the BA to use RGB-D and semantic labels in the optimization directly, and provides a generalization for the photometric loss function commonly used in direct methods. RKHS-BA can incorporate appearance and semantic labels within a hierarchical semantic-geometric functional representation that is continuous and does not require optimization via image pyramids. We develop an odometry method using local RKHS-BA graphs that, compared to existing direct odometry methods, RKHS-BA shows highly robust performance in extremely challenging scenes and the best trade-off of generalization and accuracy across extensive experiments.

5.2 Introduction

Bundle Adjustment (BA) is widely used in visual perception algorithms such as Simultaneous Localization and Mapping (SLAM) and 3D Reconstruction. It jointly optimizes visual structures and all the camera parameters to construct a spatially consistent 3D world model [184]. Existing BA methods include feature-based methods [184] and direct methods [125, 96, 59], and both are formulated as robust non-linear optimization problems.

Feature-based BA methods require extractions of sparse geometric representations, including points, lines, and planes, which are usually invariant to illumination noise or rotations [46, 45, 121, 80]. Then, in the optimization step, they minimize reprojected geometric residuals for features observed across multiple frames via multi-view geometry [184, 80]. The construction of such reprojected residuals naturally leads to sparse Hessian structures but relies on correct feature correspondences across multiple frames. Many works have been devoted to improving their robustness, such as improving frontend feature matching's quality with deep networks [73], adopting robust loss functions [184], or probabilistically modeling data association hypothesis in the backend [127, 50, 52]. However, in highly texture-less or semi-static environments, feature association contaminated with outliers is still an open problem [145].

Direct or photometric BA methods take denser representations from images, such as the edges [59], surfaces [224, 205], or the raw pixel values [125], and then optimize the photometric loss under the assumption of brightness constancy [59, 158]. With the capability of adjusting the projective association during optimization [59], photometric BA demonstrates more robustness in environments with fewer textures or more repetitive patterns. However, full images need to be stored in the pose graph even in semi-dense approaches [232]. Furthermore, their illumination invariance presumption is seriously violated in outdoor situations where complex illumination, changeable weather, and dynamic objects exist.

Rich semantic information from modern vision sensors can contribute to the robustness of BA in such challenging scenarios. Specifically, we denote various types of visual information, including pixel classes, object instances, intensities, or colors, which are invariant to pose changes, as *hierarchical semantics*. For example, dense SLAM systems such as ElasticFusion and BAD-SLAM incorporate color consistency residuals as invariant visual information in their backend optimization [202, 158]. Object detection neural networks can provide another type of semantic information, that is, 2D or 3D object proposals from image streams [72, 180]. They enable the representation of object-level entities in the factor graph for feature-based systems [152, 52, 128]. Suma++ [27] leverages point-level dense semantics in LiDAR SLAM, where point-wise semantic similarity contributes to the residual weighting.

In the chapter, we report a novel direct BA framework with semi-dense hierarchical semantic representation in a Reproducible Kernel Hilbert Space (RKHS) (shown in Figure 19). The proposed RKHS-BA constructs a specialized pose graph. Its nodes represent continuous functional representations. Its edges represent the corresponding frames' geometric and hierarchical semantic alignment. Inspired by [223], we relax the strict data correspondence in previous BA methods by associating each point observation of one frame to multiple semantically similar points in other frames. The soft association naturally arises from a functional representation in some RKHS and constrains the geometric distance and the visual similarity when a new frame is observed. The optimization stage approximates the formulation with an Iteratively Reweighted Least Square (IRLS) solver.

In particular, this work has the following contributions.

1. A novel correspondence-free direct BA framework with hierarchical semantics in an RKHS

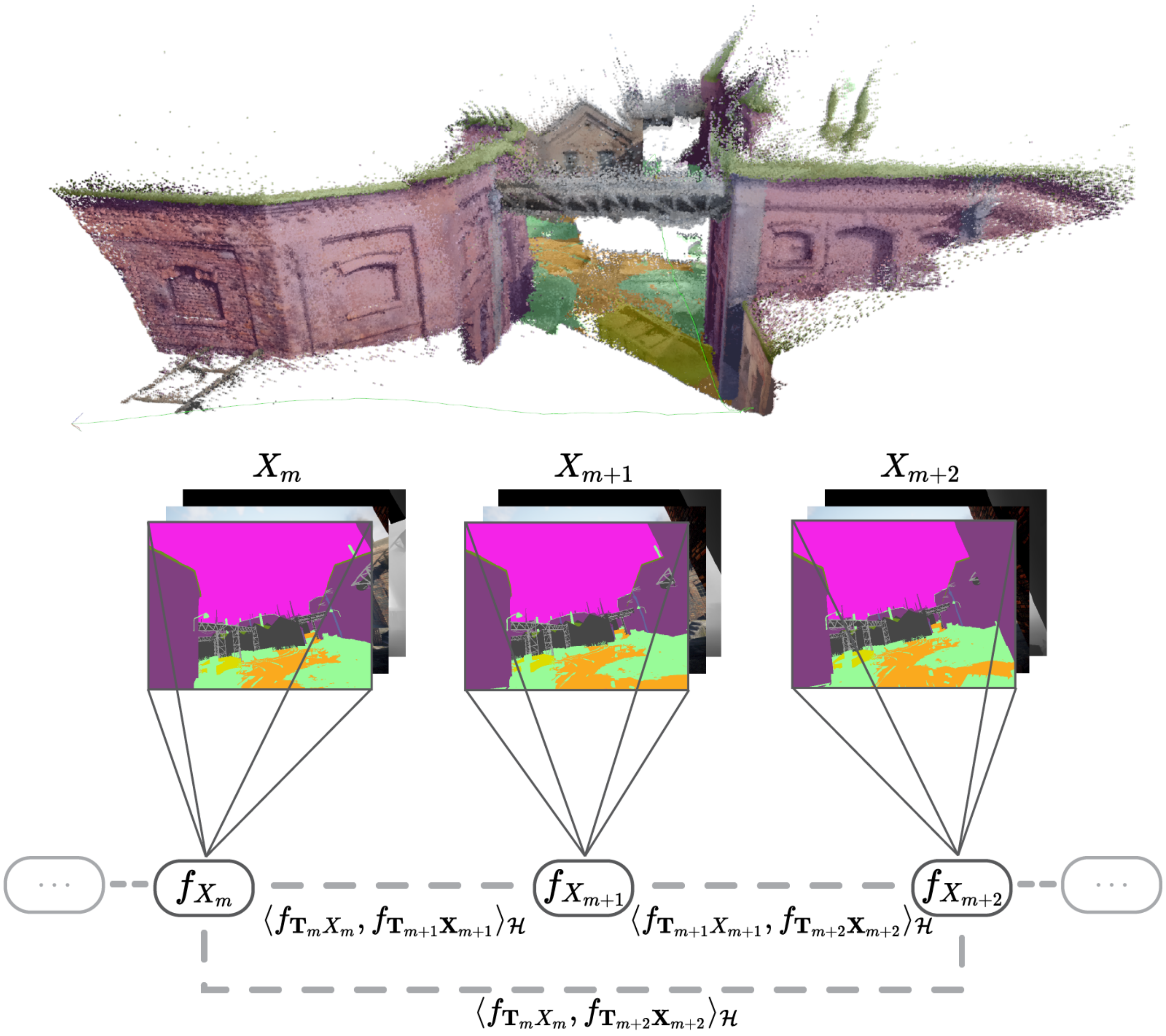

X_m
X_{m+1}
X_{m+2}
$\cdots$
f_{X_m}
$f_{X_{m+1}}$
$f_{X_{m+2}}$
$\cdots$
$\langle f_{\mathbf{T}_m X_m}, f_{\mathbf{T}_{m+1}\mathbf{X}_{m+1}} \rangle_{\mathcal{H}}$
$\langle f_{\mathbf{T}_{m+1} X_{m+1}}, f_{\mathbf{T}_{m+2}\mathbf{X}_{m+2}} \rangle_{\mathcal{H}}$
$\langle f_{\mathbf{T}_m X_m}, f_{\mathbf{T}_{m+2}\mathbf{X}_{m+2}} \rangle_{\mathcal{H}}$

Figure 19: We represent a point cloud observation as a function in the Reproducing Kernel Hilbert Space (RKHS), denoted as f_{X_m}, where X_m is the raw sensor measurements containing both geometric information like 3D points and non-geometric information such as color, intensity, and semantics. An inner product $\langle f_{\mathbf{T}_m X_m}, f_{\mathbf{T}_n X_n} \rangle_{\mathcal{H}}$ measures the alignment of two functions at timestamp m and n. The full objective function consisting of multiple frames is formulated as the sum of all inner products between all pairs of relevant frames.

representation called RKHS-BA.

2. We propose a new backend formulation of the pose graph that does not rely on strict data correspondence and encodes hierarchical semantic information in optimization.

3. We validate the proposed RKHS-BA with point cloud registration and odometry baselines on multiple synthetic and real-world datasets, including Stanford 3D Scanning Dataset [186], TUM

RGB-D Dataset [167], and TartanAir Dataset [194].

4. We provide an open-source C++ implementation.

Extension: The preliminary two-frame registration version of this work has been published in the conference ICRA 2021 [223]. Compared with the conference version, we add a series of extensions, including a new formulation of multi-frame instead of two-frame inputs, a newly developed IRLS-based solver, and experiments of various sensors from simulated or real datasets. As demonstrated by extensive evaluations, our method achieves lower pose drifts with fewer running iterations and the additional ability of largescale global BA compared to the conference version.

We denote the sequential K frames' robot poses as $\mathcal{T} = \{\mathbf{T}_1, \mathbf{T}_2, ..., \mathbf{T}_K\}$ ($\mathbf{T}_I \in \mathrm{SE}(3)$) and sensor observations $\mathcal{X} = \{X_1, X_2, ..., X_K\}$ at each timestamp. Each sensor observation contains a finite collection of homogeneous points, $X_m = \{\mathbf{x}_1^m, \mathbf{x}_2^m, ...\}$ ($\mathbf{x}_i^m \in \mathbb{R}^3$). In addition to the geometric information, every point $\mathbf{x}_i^m$ also contains pose-invariant visual information of *various* dimensions, such as color, intensity, or semantic classes. SemanticCVO [223]

Let $(V_1, V_2, \dots)$ be different inner product spaces describing different types of non-geometric features of a point, such as color, intensity, and semantics. To combine these features of different dimensions into a unified hierarchical semantic representation $\ell_{\mathcal{X}} : \mathcal{X} \to \mathcal{I}$ that is transformation-invariant, we use their tensor product, $V_1 \otimes V_2 \otimes \dots$, which also lies in an inner product space $(\mathcal{I}, \langle \cdot, \cdot \rangle_{\mathcal{I}})$ [223]. For example, for any $\mathbf{x}_i^m \in X_m$ with a 3-dimensional color feature $v_1 \in V_1$ and a 10-dimensional semantic feature $v_2 \in V_2$, its hierarchical semantic feature is $\ell_{\mathcal{X}}(\mathbf{x}_i^m) = v_1 \otimes v_2 \in V_1 \otimes V_2$.

Similar to CVO [34, 223], we represent the point cloud observations X_m at frame m into a function $f_{X_m} : \mathbb{R}^3 \to \mathcal{I}$ living in a RKHS $f_{X_m} \in (\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$. The transformation $\mathbf{T}_m$ at the corresponding timestamp m, $\mathrm{SE}(3) \curvearrowright \mathbb{R}^3$ induces an action $\mathrm{SE}(3) \curvearrowright \mathcal{H}$ by $\mathbf{T}_m f(X_m) := f(\mathbf{T}_m X_m)$. With this observation, we denote $\mathbf{T}_m f(X_m) := f_{\mathbf{T}_m X_m}$, representing the point cloud function under the transformation. In SemanticCVO [223], the inner product of the two functions at two timestamps, $\langle f_{\mathbf{T}_m X_m}, f_{\mathbf{T}_n X_n} \rangle_{\mathcal{H}}$, measures the distance in RKHS.

5.3 Problem Setup of RKHS-BA

Figure 19 shows a pose graph of multiple frames. Define $\mathcal{C}$ as the set of connected frame index pairs, and we have the full objective function over the pose graph as

$$F(\mathcal{T}) := \sum_{(m,n)\in\mathcal{C}} \underbrace{\langle f_{\mathbf{T}_m X_m}, f_{\mathbf{T}_n X_n} \rangle_{\mathcal{H}}}_{F^{mn}} \tag{5.1}$$

$$\mathcal{T}^* = \underset{\mathcal{T}}{\arg\max}\ F(\mathcal{T}), \tag{5.2}$$

We apply the similar kernel trick [223] to the information inner product. This allows us to turn the point cloud to functions via

$$f_{\mathbf{T}X}(\cdot) := \sum_{\mathbf{x}_i \in X} \ell_X(\mathbf{x}_i) k(\cdot, \mathbf{T}\mathbf{x}_i), \tag{5.3}$$

where $\ell_X(\mathbf{x}_i)$ encodes the hierarchical semantic information that does not vary with respect to robot poses. $k(\cdot, \mathbf{x}_i)$ encodes the geometric information that varies with robot poses. f_{TX} becomes a unified representation for the point cloud observation. We can now obtain the inner product of f_X and f_Z as

$$F^{mn} := \sum_{\mathbf{x}_i \in X_m, \mathbf{z}_j \in Z_n} \langle \ell_X(\mathbf{x}_i), \ell_Z(\mathbf{z}_j) \rangle_{\mathcal{I}} \cdot k(\mathbf{T}_m \mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n), \tag{5.4}$$

where

$$\begin{aligned} \langle \ell_X(\mathbf{x}), \ell_Z(\mathbf{z}) \rangle_{\mathcal{I}} &= \langle u_1 \otimes u_2 \otimes \ldots, v_1 \otimes v_2 \otimes \ldots \rangle \\ &= \langle u_1, v_1 \rangle \cdot \langle u_2, v_2 \rangle \cdot \ldots . \end{aligned} \tag{5.5}$$

By substituting (5.5) into (5.4), we obtain

$$\begin{aligned} \langle f_{\mathbf{T}_m X_m}, f_{\mathbf{T}_n Z_n} \rangle_{\mathcal{H}} &= \sum_{\substack{\mathbf{x}_i^m \in X_m \\ \mathbf{z}_j^n \in Z_n}} \big(\prod_k \langle u_{ki}, v_{kj} \rangle\big) \cdot k(\mathbf{T}_m \mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n) \\ &:= \sum_{\mathbf{x}_i^m \in X_m, \mathbf{z}_j^n \in X_n} c_{ij} \cdot k(\mathbf{T}_m \mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n). \end{aligned} \tag{5.6}$$

This inner product between the two functions above is a double sum of all pairs of points from the two point clouds. As most pairs of the points are far away, only a small subset of similar point pairs, both geometrically and hierarchical-semantically, would remain. In this way, (5.6) can be interpreted as a point-wise *soft data association* function, which considers both the geometry and the hierarchical semantics. If the current estimates of the poses change, the association will reflect the change accordingly.

Based on the above definition, the generalized objective function of RKHS-based bundle adjustment becomes

$$\begin{aligned} F(\mathcal{T}) &:= \sum_{(m,n) \in \mathcal{C}} \sum_{\mathbf{x}_i^m \in X_m, \mathbf{z}_j^n \in Z_n} \underbrace{k(\mathbf{T}_m \mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n) \cdot c_{ij}^{mn}}_{F_{ij}^{mn}} \\ \mathcal{T}^* &= \arg\max_{\mathcal{T}} F(\mathcal{T}). \end{aligned} \tag{5.7}$$

Problem (5.7) describes the full geometric and hierarchical semantic relationship for all the edges in the pose graph. Each c_{ij}^{mn} is invariant to the relative transformation; thus, it will be a constant during optimization. In our implementation, the double sum in (5.6) is sparse because a point $\mathbf{x}_i \in X$ is far away from the majority of the points $\mathbf{z}_j \in Z$, either in the spatial (geometry) space or one of the feature (semantic) spaces. The sparsity is dependent on the level of the feature space differences.

As a special case, if there are only two frames, the inner product formulation in (5.7) reduces to the RKHS-registration in [223]. If the hierarchical semantic information is not used, the alignment of two geometric point clouds reduces to Kernel Correlation [185]. The two-frame case can be solved by gradient ascent in the optimization [34].

5.4 Semantically Informed Iteratively Reweighted Least Squares Backend

In this section, we propose approximating the objective function in (5.7) for the general multi-frame case with the IRLS approach. The special case of the two-frame registration adopts gradient ascent for optimizing the objective function in (5.6) [223]. Aiming at an efficient convergence in a potentially large-scale multi-frame situation, we choose IRLS by design.

5.4.1 From RKHS to Semantically Weighted Least Squares

For the kernel of our RKHS, $\mathcal{H}$, we choose the squared exponential kernel $k : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$:

$$k(\mathbf{x}, \mathbf{z}) = \sigma^2 \exp\left(\frac{-\|\mathbf{x} - \mathbf{z}\|_3^2}{2\ell^2}\right), \tag{5.8}$$

for some fixed real parameters (hyperparameters) σ and ℓ (the *lengthscale*), and $\|\cdot\|_3$ is the standard Euclidean norm on $\mathbb{R}^3$. With a good initialization of the frame poses $\mathcal{T} = \{\mathbf{T}_1, ..., \mathbf{T}_K\}$ from tracking, and let $d(\mathbf{x}, \mathbf{z}) := \mathbf{x} - \mathbf{z}$, we can expand each term F_{ij}^{mn} in (5.1)

$$\begin{aligned} F_{ij}^{mn} &= k(\mathbf{T}_m \mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n) \cdot c_{ij}^{mn} \\ &= c_{ij}^{mn} \sigma^2 \exp\left(\frac{-\|\mathbf{T}_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n\|_3^2}{2\ell^2}\right) \\ &:= c_{ij}^{mn} k(d(\mathbf{T}_m \mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2) \end{aligned} \tag{5.9}$$

If we apply a perturbation $\boldsymbol{\epsilon}_m \in \mathbb{R}^6$ on the right of $\mathbf{T}_m$ as

$$\mathbf{T}_m^\star = \mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge) = \mathbf{T}_m \exp(\begin{bmatrix} \rho_m \\ \phi_m \end{bmatrix}^\wedge). \tag{5.10}$$

Then the gradient with respect to $\boldsymbol{\epsilon}_m$ is

$$\begin{aligned} \nabla F_{ij}^{mn} &= c_{ij}^{mn} \frac{\partial k(d(\mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge)\mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2)}{\partial d} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} \\ &= c_{ij}^{mn} \frac{\partial k(d(\mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge)\mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2)}{\partial d} \frac{1}{d} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} d \\ &= c_{ij}^{mn} k \frac{-2d}{2\ell^2} \frac{1}{d} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} d \\ &= \frac{-1}{\ell^2} \underbrace{c_{ij}^{mn} k}_{w_{ij}^{mn}} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} d, \end{aligned} \tag{5.11}$$

where we denote the term

$$w_{ij}^{mn} := c_{ij}^{mn} k(d(\mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge)\mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2).$$

After summing it up for all pairs of $(m, n) \in \mathcal{C}$ and $\mathbf{x}_i^m \in X_m$, $\mathbf{z}_j^n \in Z_n$ and taking the gradients to zero, we obtain

$$\sum_{(m,n)\in\mathcal{C}} \sum_{\substack{\mathbf{x}_i^m \in X_m \\ \mathbf{z}_j^n \in Z_n}} w_{ij}^{mn} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} d = 0. \tag{5.12}$$

Here, the weight w_{ij}^{mn} encodes the full geometric and hierarchical semantic relations between the pair of points. In real data, a point's color or semantic features can differ from most other points. Thus, the weight will effectively suppress the originally dense residuals between this point and all the other points. If we treat w_{ij}^{mn} as *constant* weights during one optimization step, the solution to (5.12) corresponds to the solution for the following least squares problem:

$$\arg\max_{\mathcal{T}} \sum_{(m,n)\in\mathcal{C}} \sum_{\mathbf{x}_i^m \in X_m, \mathbf{z}_j^n \in Z_n} w_{ij}^{mn} d(\mathbf{T}_m \mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2, \tag{5.13}$$

where $\mathcal{T}$ are the poses of all the keyframes involved except the first frame. To see that, we can apply the perturbation $\exp(\boldsymbol{\epsilon}_m^\wedge)$ on the right of $\mathbf{T}_m$ and then take the gradient with respect to $\boldsymbol{\epsilon}_m$ for (5.13). During the optimization, the weight value w_{ij}^{mn} is re-calculated after every step update due to the pose changes.

Problem (5.13) is a nonlinear least squares on the $\mathrm{SE}(3)$ manifold that can be solved with an IRLS algorithm [165, 187, 85, 218]. It can be solved with an off-the-shelf solver like [3]. Please refer to the Appendix for the detailed derivation.

5.4.2 Lengthscale Decay

In classical featured-based and photometric bundle adjustments, residuals are collected from image pyramids to consider feature points at different scales [121, 60]. In RKHS, the point clouds are represented with continuous functions, where the lengthscale ℓ of the geometric kernel in (5.8) controls the scale [223]. Starting with a large value for a global perspective, the lengthscale is reduced by a fixed percentage for finer-grain registration until the number of non-zero terms in (5.13) stabilizes. Each pair of frames decays its lengthscale independently.

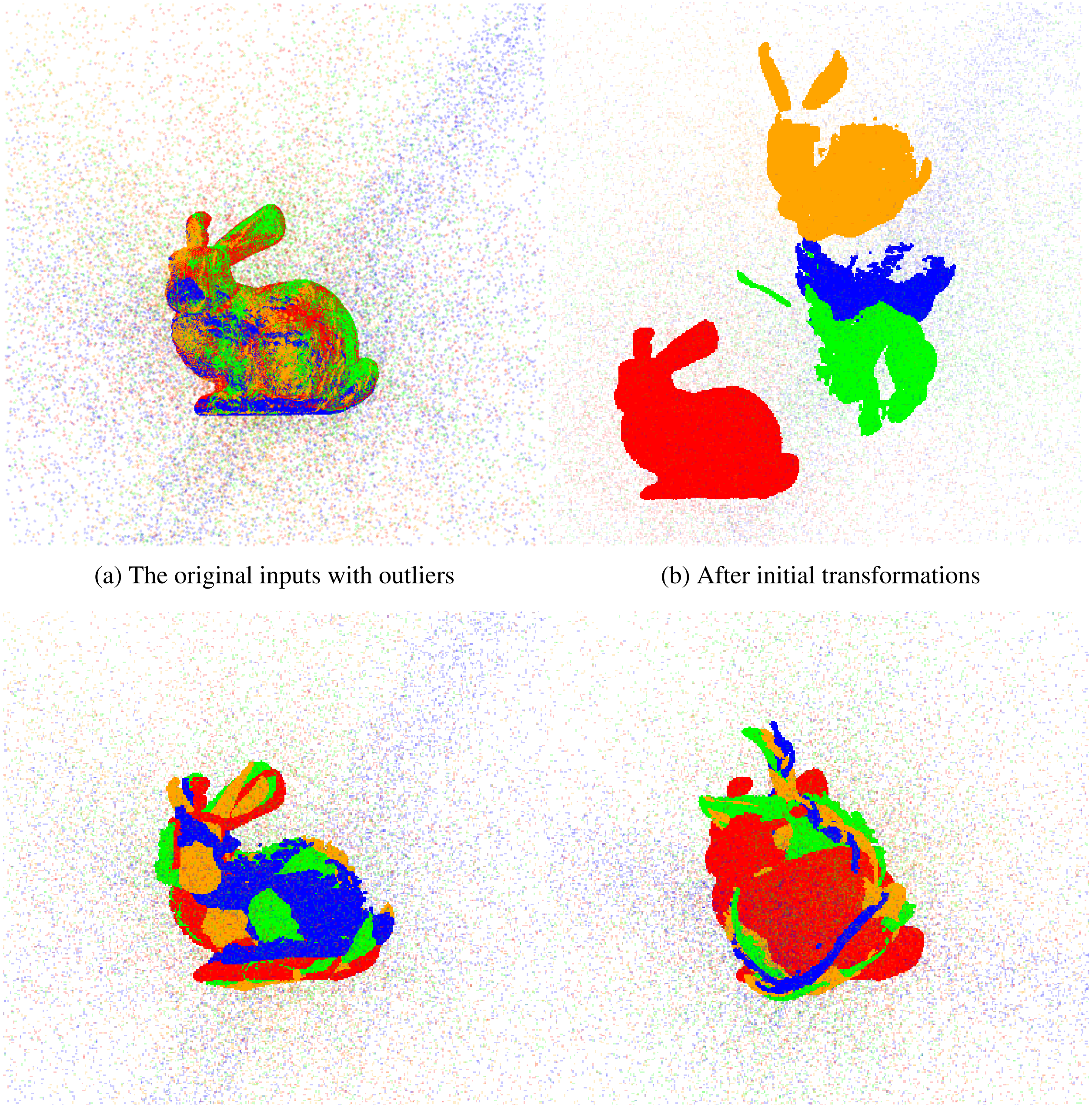
5.5 Experimental Results

We evaluate the proposed method with multi-frame registration and visual odometry experiments in publicly available datasets. We start with toy examples of four-frame registrations on partially overlapped geometric and semantic point clouds. The motivation is to stress-test the proposed method's performance under different initialization and outlier ratios. Next, we present indoor and outdoor experiments in structured, textureless, and semi-static environments to test the performance in real applications. The depth sources come from RGB-D cameras and neural network predictions. We run the experiments on a desktop with a 48-core Intel(R) Xeon(R) Platinum 8160 CPU and an Nvidia Titan RTX GPU.

5.5.1 Simulated Example of Multi-point cloud registration

We present two toy examples of multi-frame registration on the Stanford Bunny dataset [186], shown in Figure 20, and the TartanAir dataset [194], shown in Figure 21. The Bunny Dataset provides only geometric point clouds. The TartanAir Dataset provides color and semantic point clouds. We chose four scans that do not completely overlap. They are further downsampled with a voxel filter.

The four point clouds are initialized as follows. First, they are randomly rotated with four different angles, 12.5°, 25°, 37.5°, and 50°, along a random axis. Second, random translations are further applied. Third, we perturb the point clouds with five different outlier ratios: $0\%, 12.5\%, 25\%, 37.5\%$, and 50%. A perturbation is added in the normal direction of every point.



(a) The original inputs with outliers

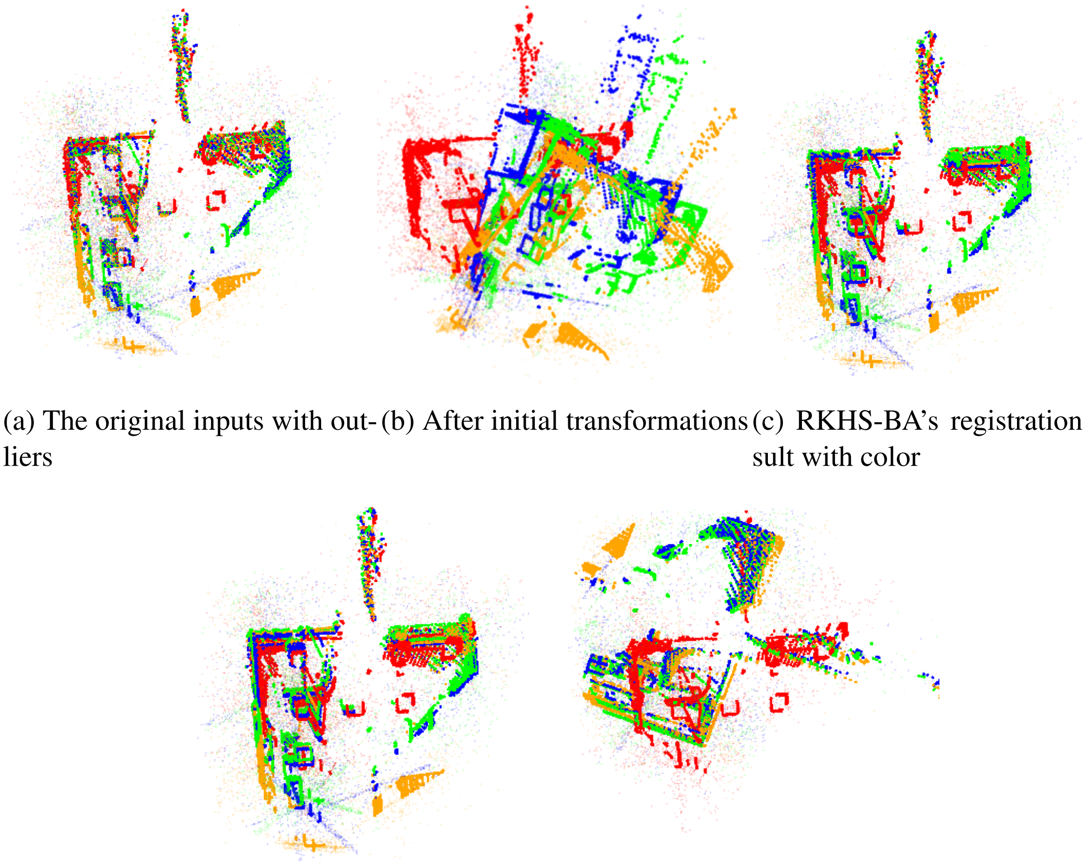
(b) After initial transformations

(c) RKHS-BA's registration result

(d) JRMPC's registration result

Figure 20: An example of a four-view point cloud registration test with only geometric information on the Bunny [186] Dataset. (a) The four partially overlapped point clouds of the Bunny Dataset, each perturbed by 50% random outliers. (b) The four Bunny point clouds after we apply initial rotations of 50 degrees around random axes and a random translation of $0.5m$. (c) RKHS-BA's registration result. (d) JRMPC's [61] registration result.$\gamma = 0.1$.

If a point is an outlier, a uniformly sampled noise is added in the specified interval around the point. Otherwise, we add a Gaussian noise centered around the point's original position. We generate 40



(a) The original inputs with outliers

(b) After initial transformations

(c) RKHS-BA's registration result with color

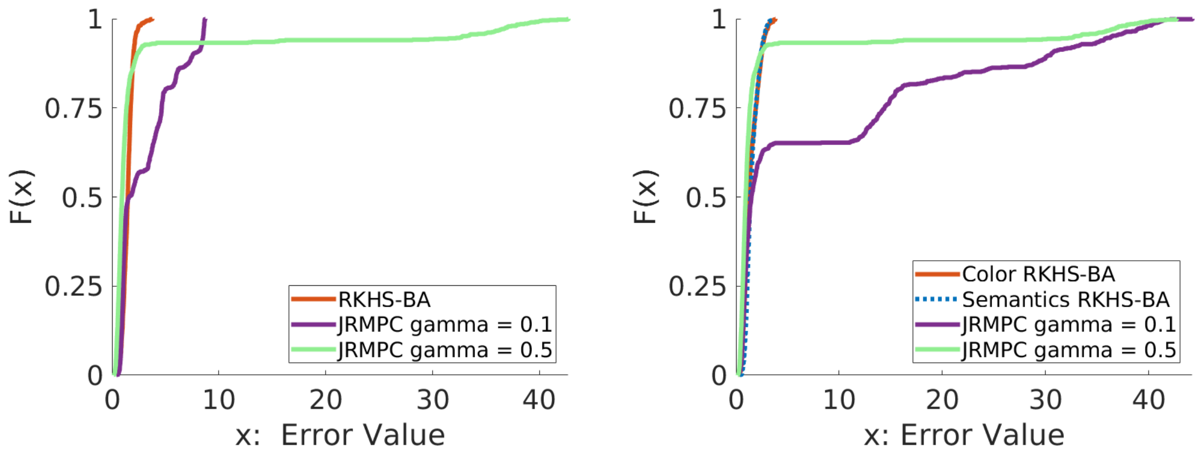
(d) RKHS-BA's registration result with color and semantic labels

(e) JRMPC's registration result

Figure 21: An example of a four-view point cloud registration test on TartanAir [194] `Hospital-Easy-P001` sequence. The four point clouds are sampled every 20 frames. The semantic labels for every frame are provided by the dataset. (a) The initial four different frames of the TartanAir Dataset, each perturbed by 50% random outliers. (b) The four Tartanair point clouds after we apply initial rotations of 50 degrees around random axes and a random translation of $4m$. (c) RKHS-BA's registration result with only color information. (d) RKHS-BA's registration result with both color and semantic labels. (e) JRMPC's [61] registration result with $\gamma = 0.1$.

random initializations for each angle and outlier ratio pair above.

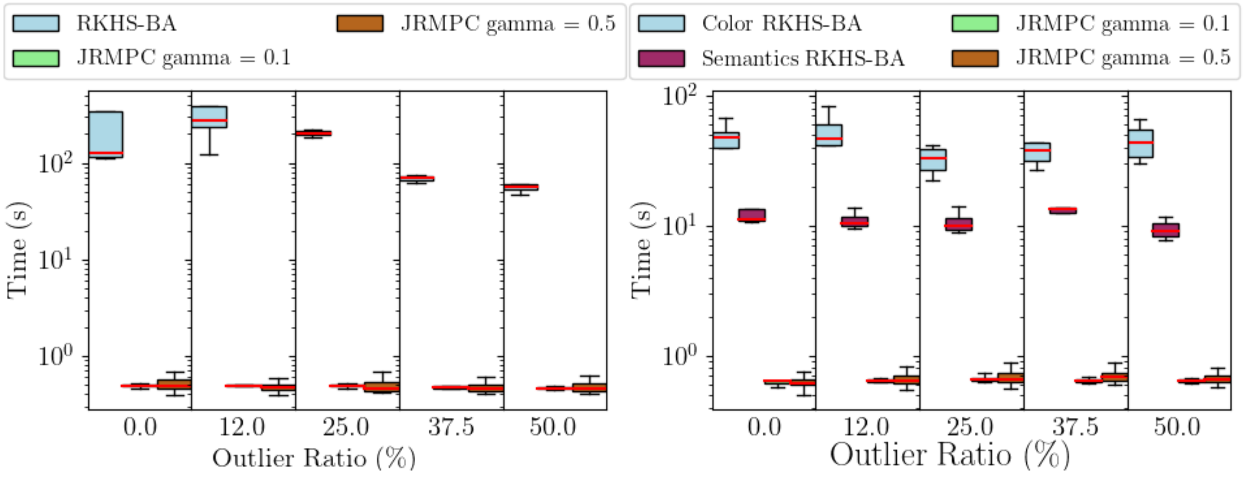
We compare our registration results with JRMPC [61], which is a multi-frame geometric registration baseline based on the Gaussian Mixture Model (GMM). We evaluate a single registration result with the sum of Frobenius Norm (denoted as $\|\cdot\|_{\mathrm{F}}$) of the errors of the other three frames'



(a) CDF for Bunny registration test

(b) CDF for TartanAir registration test

Figure 22: The error CDF plot of all the four-view point cloud registration tests on the Bunny [186] and TartanAir [194] Dataset (a) The error CDF for all the Bunny experiments. (b) The error CDF for all the TartanAir experiments.



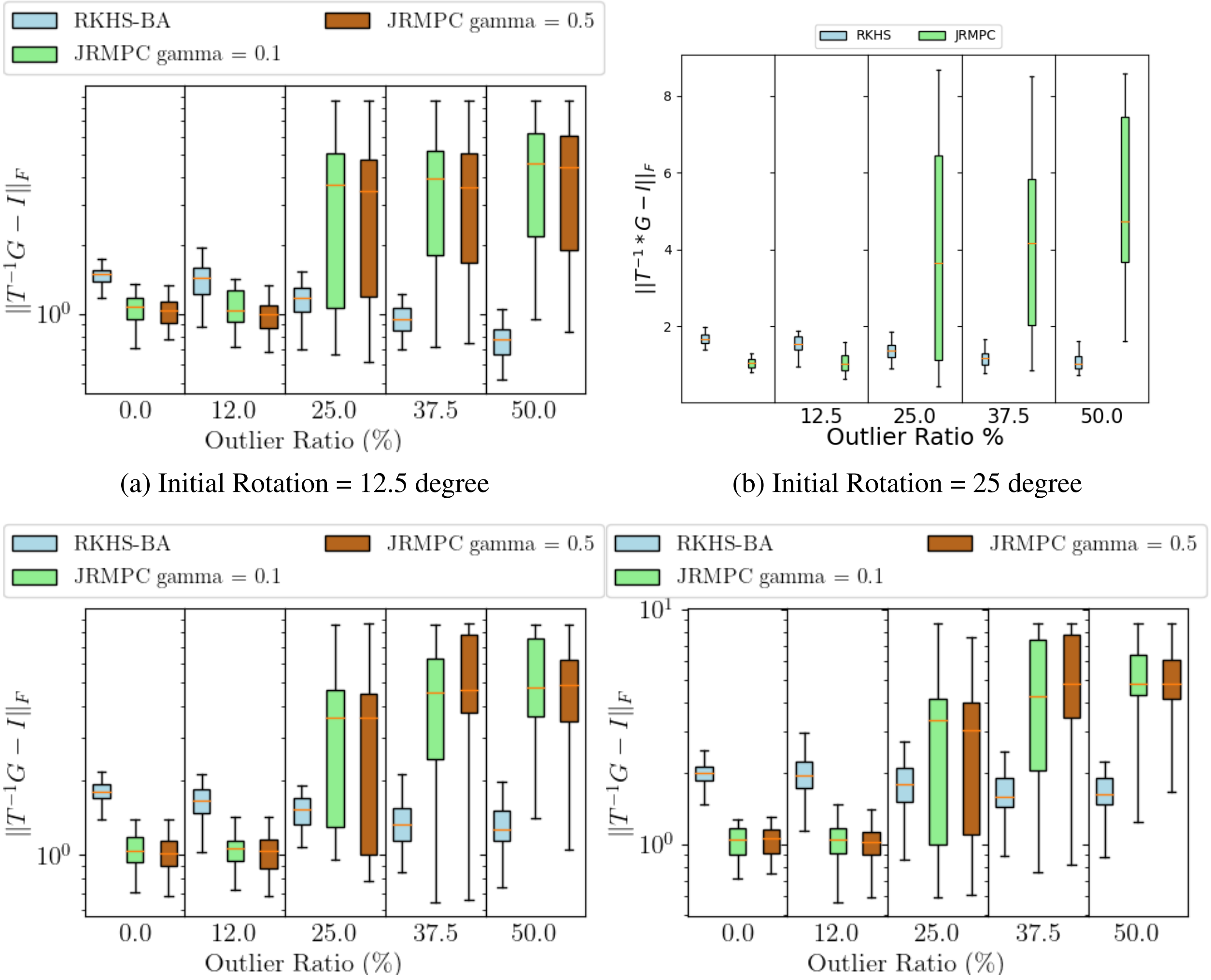
(a) Time for Bunny registration test

(b) Time for TartanAir registration test

Figure 23: The running time statistics for a single four-view registration of all the experiments. (a) Box Plot for the registration time on the Bunny Dataset [186] (b) Box Plot for the registration time on the TartanAir Dataset [194]

poses with respect to the first frame,

$$\sum_{i=2}^{4} \|\mathbf{T}_i^{-1}\mathbf{G}_i^{(\mathrm{gt})} - \mathbf{I}\|_{\mathrm{F}}.$$



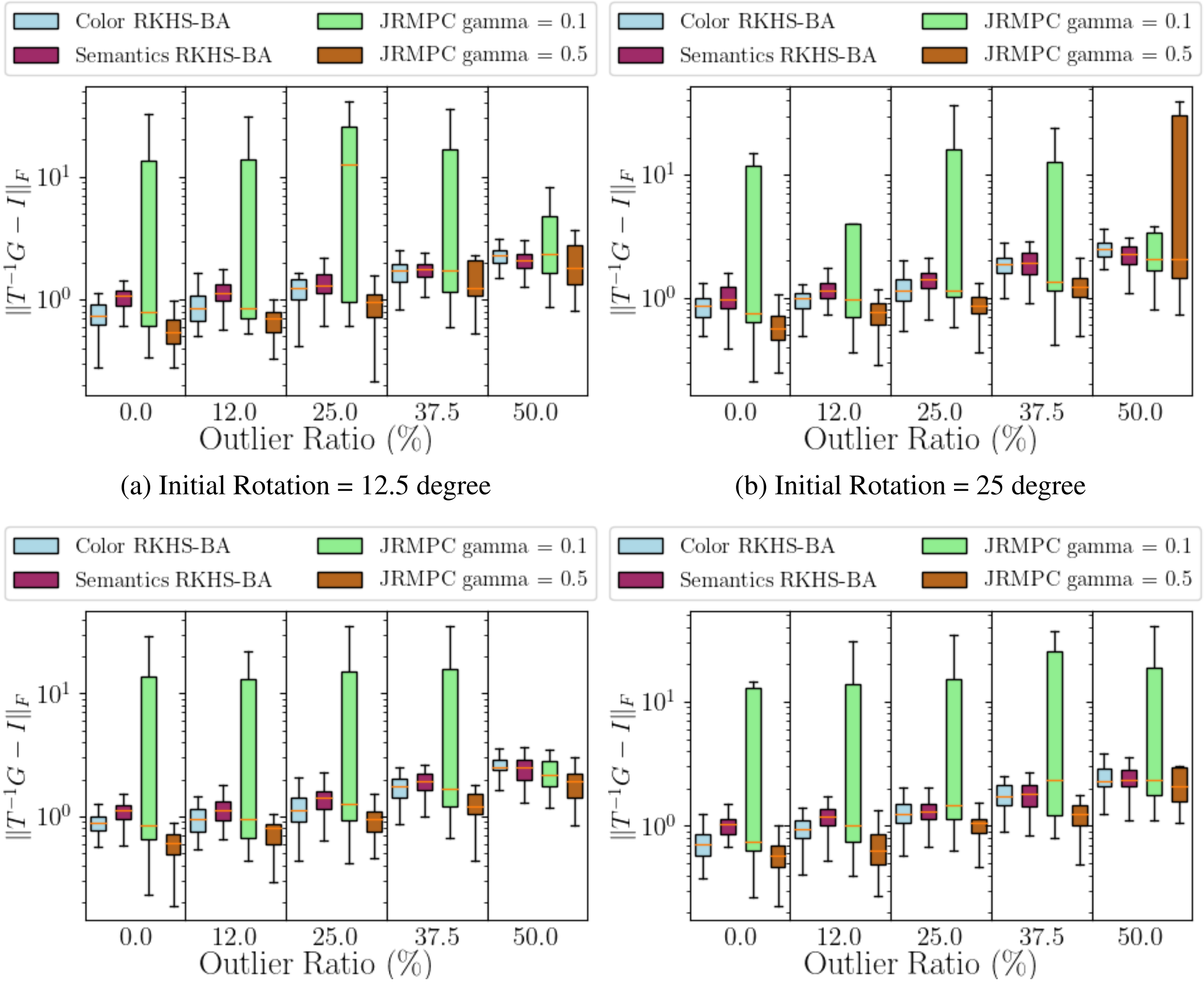
(a) Initial Rotation = 12.5 degree

(b) Initial Rotation = 25 degree

(c) Initial Rotation = 37.5 degree

(d) Initial Rotation = 50 degree

Figure 24: The benchmark results of the four-frame registration tests on the Bunny Dataset [186]. Each box plot contains the resulting pose errors in the Frobenius Norm of different outlier ratios at the same initial rotation angle. (a) The initial angle is 12.5 degrees. (b) The initial angle is 25 degrees. (c) The initial angle is 37.5 degrees. (d) The initial angle is 50 degrees.



(a) Initial Rotation = 12.5 degree

(b) Initial Rotation = 25 degree

(c) Initial Rotation = 37.5 degree

(d) Initial Rotation = 50 degree

Figure 25: The benchmark results of the four-frame registration test on the TartanAir Dataset [194]. We include both Color RKHS-BA, which takes color information, and Semantic RKHS-BA, which takes both color and semantic labels. Each box plot contains the resulting pose errors in the Frobenius Norm of different outlier ratios at the same initial rotation angle. (a) The initial angle is 12.5 degrees. (b) The initial angle is 25 degrees. (c) The initial angle is 37.5 degrees. (d) The initial angle is 50 degrees.

where $\mathbf{G}_i^{(\mathrm{gt})} \in SE(3)$ is the ground truth pose.

5.5.1.1 Multi-Point Cloud geometric registration

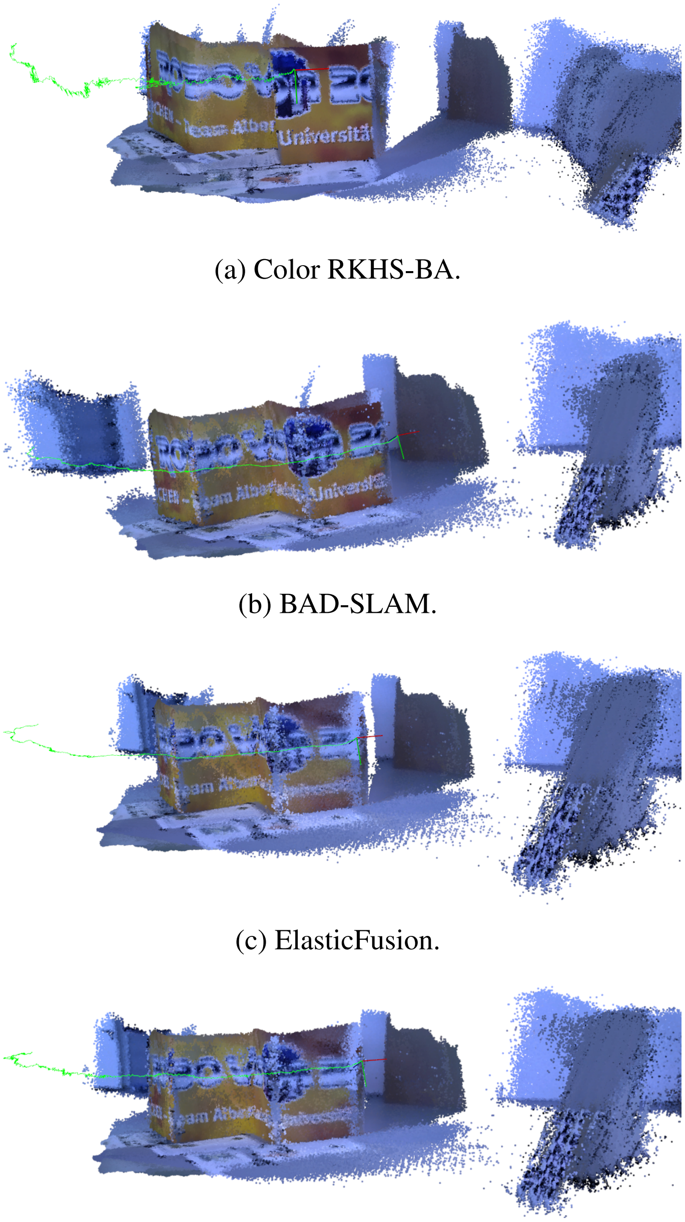
In the Bunny dataset [186], we choose four frames that are not fully overlapped from the original scan. The norms of the random initial translations are less than $1m$. The uniform noise for every outlier point is randomly sampled from the $[-0.5m, +0.5m]$ interval. The Gaussian noise for every inlier point is centered around the point's original position with a standard deviation of $0.01m$. In this experiment, we also select two different outlier ratio parameter setups for JRMPC, denoted as γ in its chapter. γ is a positive scalar specifying the proportion of outliers used to calculate the prior distribution in JRMPC.

We report the results for every outlier ratio and initial angle pair with box plots in Figure 24 and the error cumulative distribution function (CDF) plot in Figure 22a. JRMPC has slightly lower errors when the outlier ratio is small but is not robust when the outlier ratio grows above 25%. RKHS-BA is not sensitive to a larger outlier ratio. It can achieve consistently low errors in most of the experiment cases. In this experiment, a larger outlier ratio ($\gamma = 0.5$) of JRMPC has slightly better performance than $\gamma = 0.1$. The error CDF plot in Figure 22a also shows that the baseline has more failed cases than the proposed method. The result of the Bunny registration experiment is visualized in Figure 20. We are able to achieve smaller errors compared to JRMPC.

5.5.1.2 Multi-Point Cloud color and semantic registration

In the TartanAir dataset [194], we choose four frames from the `Hospital-Easy-P001` indoor sequence. The four point clouds are sampled every 20 frames. The norms of the random initial translations are less than $4m$. The uniform noise for every outlier point is randomly sampled from the $[-4m, +4m]$ interval. The Gaussian noise for every inlier point is centered around the point's original position with a standard deviation of $0.4m$. We also use the same outlier ratio parameter setups for JRMPC as in the Bunny Experiment.

As shown in Figure 25, the Color and Semantic RKHS-BA have similar errors under different initial rotations and outlier rates. JRMPC is sensitive to the choice of the outlier ratio parameter γ. It has significantly larger errors at all the initial values when $\gamma = 0.1$. It has lower errors at larger actual outlier rates (37.5% and 50%) but is also not robust when the actual outlier rate is 25%. According to the CDF plot in Figure 22b, when $\gamma = 0.1$, JRMPC achieves better performance than the case when $\gamma = 0.5$, but it still has more failed cases than our method. The result of the TartanAir registration experiment is visualized in Figure 21. We can achieve small errors even when the outlier ratio is very large.



(a) Color RKHS-BA.

(b) BAD-SLAM.

(c) ElasticFusion.

(d) ORB-SLAM2.

Figure 26: Qualitative comparisons of the stacked point cloud map of the four methods above in the TUM RGB-D `fr3-structure-texture` sequence [167]. Color RKHS-BA in (a) shows clearer surface texture on both the poster and the chess board than other methods. BAD-SLAM [158] in (b) succeeds in reconstructing the texture on the surface that is not directly facing the camera. ElasticFusion [202] in (c) and ORB-SLAM2 [121] in (d) get similar results, and the texture is more blurry than other methods.

5.5.1.3 Time Analysis

Assuming there are M edges in the pose graph and each frame has $O(N)$ points, then the time complexity would be $O(MN^2)$ because of the cost to evaluate all pairs of inner product values. In practice, we approximate the full inner products by only considering pairs of points with small Euclidean distances. This step is implemented with a 3D K-Nearest-Neighbor Search [119], which reduces the time complexity to $O(MN \log N)$.

The time consumption in the four-frame registration tests is listed in Figure 23. JRMPC is significantly faster in all the examples. Interestingly, the additional hierarchical semantic information improves RKHS-BA's running speed because it helps sparsify the number of nontrivial inner products.

5.5.2 Application: Sliding Window Bundle Adjustment of RGB-D inputs

We evaluate the proposed BA algorithm on multiple sequences of the RGB-D Dataset [167] and TartanAir Dataset [194], including experiments in static and structured, textureless, and semi-static environments. We present quantitative evaluations of the trajectories as well as qualitative comparisons of the stacked point cloud maps versus the mainstream algorithms.

5.5.2.1 Baseline setup

In the experiments, we implement the proposed formulation into a frontend and a backend. The frontend is a special case of the inner product in (5.1) of two frames and provides initial pose values for the backend. It takes around 2000 semi-dense points from an input image generated with DSO [59]'s point selector. The backend uses the full inner product formulation on the latest four keyframes and estimates the final poses. Both datasets use fixed sets of hyperparameters within their sequences.

We compare our approach with four visual SLAM or odometry systems: BAD-SLAM [158], ORB-SLAM2 [121], ElasticFusion [202] and StereoDSO [193]. StereoDSO is the closest baseline because of its backend's semi-dense photometric bundle adjustment. BAD-SLAM and ElasticFusion both feature a joint color and geometric optimization in the backend, although they have independent map fusion steps. With a featured-based bundle adjustment module, ORB-SLAM2 does not share the same perspective as direct SLAMs but is listed here for reference. We use BAD-SLAM, ORB-SLAM2, and ElasticFusion's officially released code with RGB-D inputs. Since StereoDSO's original implementation has not been released, we reproduced DSO's results using an open-source implementation [206], which contains DSO with stereo depth initialization. For a fair comparison, all the methods' global loop closure modules are turned off.

5.5.2.2 TUM RGB-D Dataset with Color Information

On TUM RGB-D Dataset [167], we use the depth measurements from the RGB-D sensor. We evaluate the proposed method on the `fr1` sequences containing static and structured inputs, as well as the `fr3` sequences containing structureless and textureless environments. We evaluate the resulting trajectories with Relative Pose Error (RPE) provided in the official benchmark kit.

As shown in Table 5.1 and Table 5.2, the proposed method's drifts ($0.0689\mathrm{m}/\sec$) are close to ElasticFusion's ($0.0616\mathrm{m}/\sec$) in the `fr1` office sequences, both behind ORB-SLAM2 ($0.0458\mathrm{m}/\sec$). In `fr3` sequences with less structure or texture, the proposed method has lower average drifts ($0.1017\mathrm{m}/\sec$) than the two dense methods($0.6121\mathrm{m}/\sec$ and $0.5634\mathrm{m}/\sec$). In the meantime, RKHS-BA has a smaller standard deviation. ORB-SLAM2 has smaller drifts on the successfully-completed sequences but fails on a nostructure-notexture sequence. Figure 26 presents the stacked point cloud maps based on the resulting trajectories on the `fr3-structure-texture` sequence. RKHS-BA shows clearer surface texture on both the poster and the chess board than the baselines.

5.5.2.3 TartanAir Dataset with Semantic Information

We present semantic BA results on the TartanAir dataset [194]. The TartanAir dataset contains photo-realistic simulations of environments with ground truth depth and semantic measurements. We selected sequences that included different weather conditions to demonstrate the robustness of the proposed method. The input depth images are generated with Unimatch [209] from stereo image pairs. The semantic segmentation labels provided in the dataset are raw object IDs generated by the simulator. We merge less frequent IDs into a single class, resulting in a maximum of 10 classes. In the quantitative comparison, we calculate the drift in Absolute Translation Error (ATE) in meters using the evaluation tool provided by TartanAir [194].

The quantitative results are listed in Table 5.3. The qualitative comparisons of all the methods on three challenging sequences are shown in Figure 27. The point cloud mapping results of our method and baselines in the `hospital` sequence are shown in Figure 28. RKHS-BA, which takes color point clouds, has lower mean drifts ($0.664m$) than the remaining direct methods with color or intensity inputs. RKHS-BA with both color and semantic inputs outperforms Color RKHS-BA ($0.584m$). Both demonstrate a small standard deviation in the results as well. The feature-based method still performs the best on the two well-structured sequences, `gascola` and `seasonsforest`, when it is able to complete. But in sequences with repetitive patterns, such as `hospital`, data association becomes difficult for feature-based backends. Furthermore, in sequences with dynamic weather, like the rainy `soulcity`, the images are contaminated with raindrops and water reflections. As shown in Figure 27c, even direct backends cannot do well,

Table 5.1: The RMSE of Relative Pose Error (RPE) for `fr1` sequences. The trans. columns show the RMSE of the translational drift in $\mathrm{m}/\sec$ and the rot. columns show the RMSE of the rotational error in $\deg/\sec$.

	Color RKHS		BAD-SLAM [158]		ElasticFusion [202]		ORB-SLAM2 [121]	
Sequence	RPE Trans.	RPE Rot.	RPE Trans.	RPE Rot.	RPE Trans.	RPE Rot.	RPE Trans.	RPE Rot.
fr1/desk	0.0547	3.0200	1.5757	88.5013	0.0283	1.4442	**0.0206**	**1.3656**
fr1/desk2	0.0691	3.7674	0.0740	3.3583	0.0489	2.2814	**0.0227**	**1.9116**
fr1/room	0.0676	3.1025	0.2584	7.4462	0.0600	2.8668	**0.0487**	**1.8325**
fr1/360	**0.1309**	4.6560	1.1642	38.0795	0.1460	7.9668	0.1502	**3.4316**
fr1/teddy	0.0738	2.3343	2.5763	100.0148	0.0637	1.7727	**0.0426**	**1.3805**
fr1/xyz	0.0405	1.9358	0.0225	1.4072	0.0183	0.8517	**0.0157**	**1.0039**
fr1/rpy	0.0545	3.5829	0.0396	3.3716	0.0435	2.8264	**0.0361**	**2.4827**
fr1/plant	0.0602	2.1086	0.0809	3.3150	0.0841	4.2299	**0.0178**	**1.0376**
Mean	0.0689	3.0634	0.7240	30.6867	0.0616	3.0300	**0.0458**	**1.8057**
Median	0.0709	3.0612	0.6023	22.4275	0.0545	2.5539	**0.0493**	**1.8686**
STD	**0.0272**	0.9274	0.9535	41.1324	0.0399	2.2450	0.0493	**0.8213**

Table 5.2: The RMSE of Relative Pose Error (RPE) for the `fr3` structure v.s texture sequence. The Trans. columns show the RMSE of the translational drift in $\mathrm{m}/\sec$ and the Rot. columns show the RMSE of the rotational error in $\deg/\sec$. If a method doesn't complete a sequence, a bar ("$-$") symbol is reported.

			Color RKHS		BAD-SLAM [158]		ElasticFusion [202]		ORB-SLAM2 [121]	
structure-texture			RPE Trans.	RPE Rot.	RPE Trans.	RPE Rot.	RPE Trans.	RPE Rot.	RPE Trans.	RPE Rot.
×	✓	near	0.0673	2.9208	0.1153	3.2149	0.0149	0.7662	**0.0137**	**0.7969**
×	✓	far	0.1143	1.8551	0.1454	2.2520	0.4531	19.9528	**0.0593**	**1.4011**
✓	×	near	0.0509	4.2206	0.0266	0.9841	0.2255	0.2255	**0.0252**	**1.2838**
✓	×	far	0.0480	1.5334	0.7249	2.1433	0.0219	0.9406	**0.0114**	**0.4603**
✓	✓	near	0.0615	3.1548	0.0338	1.4718	0.0126	0.7403	**0.0104**	**0.6141**
✓	✓	far	0.0390	0.9854	0.0607	1.6148	**0.0088**	**0.4112**	0.0124	0.4992
×	×	near	0.2863	9.5914	1.3621	2.8592	1.1007	56.9050	-	-
×	×	far	0.1462	1.5289	2.4284	100.9957	2.6699	36.0210	**0.0294**	**0.8794**
Mean			**0.1017**	**3.2238**	0.6121	14.4420	0.5634	19.6849	-	-
Median			**0.0644**	**2.3880**	0.1304	2.1977	0.1237	10.4467	-	-
STD			**0.0832**	**2.7839**	0.8717	34.9805	0.9302	22.6098	-	-

while the color and semantic RKHS-BA still report low translation errors.

5.5.3 Application: Lidar Global Mapping

Lidar global mapping is another application of RKHS-BA. Classical Lidar SLAM methods perform PGO after loops are detected, but PGO only considers the consistency of poses, not the consistency of the map. In contrast, when camera-based SLAMs [121] add an extra step besides PGO, that is global bundle adjustment, to enforce the constancy of the map between the frames as well.

Table 5.3: Results of the proposed frame-to-frame method using the TartanAir benchmark as evaluated on the ATE in meters. If a method doesn't complete a sequence, the frame's index with lost tracking will be recorded in the parenthesis.

			Semantic-based direct method	Intensity-based direct method				Feature-based method
Sequence (Easy P001)	Environment	No. Frames	Semantic RKHS ATE (m)	Color RKHS ATE (m)	DSO-Stereo [206] ATE (m)	BAD SLAM [158] ATE (m)	ElasticFusion [202] ATE (m)	ORB-SLAM2 [121] ATE (m)
abandonedfactory	Sunny	434	**0.3010**	0.3149	`(412)`	1.3642	8.0056	`(410)`
gascola	Foggy	382	0.0878	0.0905	5.4988	0.1893	1.7340	**0.0377**
hospital	Repetitive	480	**0.5535**	0.5675	0.9567	`(434)`	2.8675	`(238)`
seasonsforest	Forest	319	0.1399	0.1395	`(307)`	17.0627	1.7279	**0.0359**
seasonsforest winter	Snowy	847	**1.1515**	1.5631	7.4030	`(591)`	14.4673	`(582)`
soulcity	Rainy	1083	1.4628	**1.4563**	`(910)`	`(271)`	5.6583	`(480)`
seasidetown	Textureless	403	0.3901	**0.3761**	`(30)`	218.9929	4.9269	`(260)`
Mean	-	-	**0.5838**	0.6440	-	-	5.6263	-
Median	-	-	0.3901	**0.3761**	-	-	4.9269	-
STD	-	-	**0.5254**	0.6126	-	-	4.5148	-

5.5.3.1 Setup

Assuming the trajectory of Lidar PGO is given, we first construct a pose graph for RKHS-global-BA. For a frame f_i, we firstly connect its adjacent frames f_{i-1} and f_{i+1} as odometry constrains, then the frames whose translation is within a 3-meter boundary of the frame f_i as loop closing constrains. For the edges connecting adjacent frames, we assign a smaller initial lengthscale. For the non-adjacent loop closing edges, we assign a larger initial lengthscale.

In addition, due to the large number of Lidar points per frame, we downsample the input point clouds with voxel filters. To ensure that each frame has enough line points and surface points, we use $0.4m$ voxels for surfaces and $0.1m$ for lines. This ensures that each frame contains less than $10,000$ points.

We benchmark the proposed method and the baselines on the KITTI Lidar dataset. Using the same set of hyperparameters, we evaluate the proposed method on six sequences, `00, 02, 05, 06, 07, 09`, with all the loop closures selected above. The odometry poses come from MULLS's [130] pose graph optimization result. We use the official evaluation tool from KITTI's website [69], which measures the translational drift, as a percentage ($\%$), and the rotational drift in degrees per meter ($^\circ/m$) on all possible subsequences of 100, 200...., 800 meters.

5.5.3.2 Baselines

The baseline of the proposed BA formulation is the point-to-line and the point-to-plane formulations in the mainstream Lidar bundle adjustment methods. We choose BALM [106] and HBA [105] as baselines because they provide open-source implementations. Note that BALM and HBA have extra components, such as submap and hierarchical submaps, other than the optimization of the point-to-feature cost itself. We enable these additional modules for the completeness of their implementations. We also add A-LOAM [183] which is an implementation of LOAM [220], which shows the results without loop closures. The baselines also use the same initial poses from pose graph optimization and the same input point clouds as RKHS-BA.

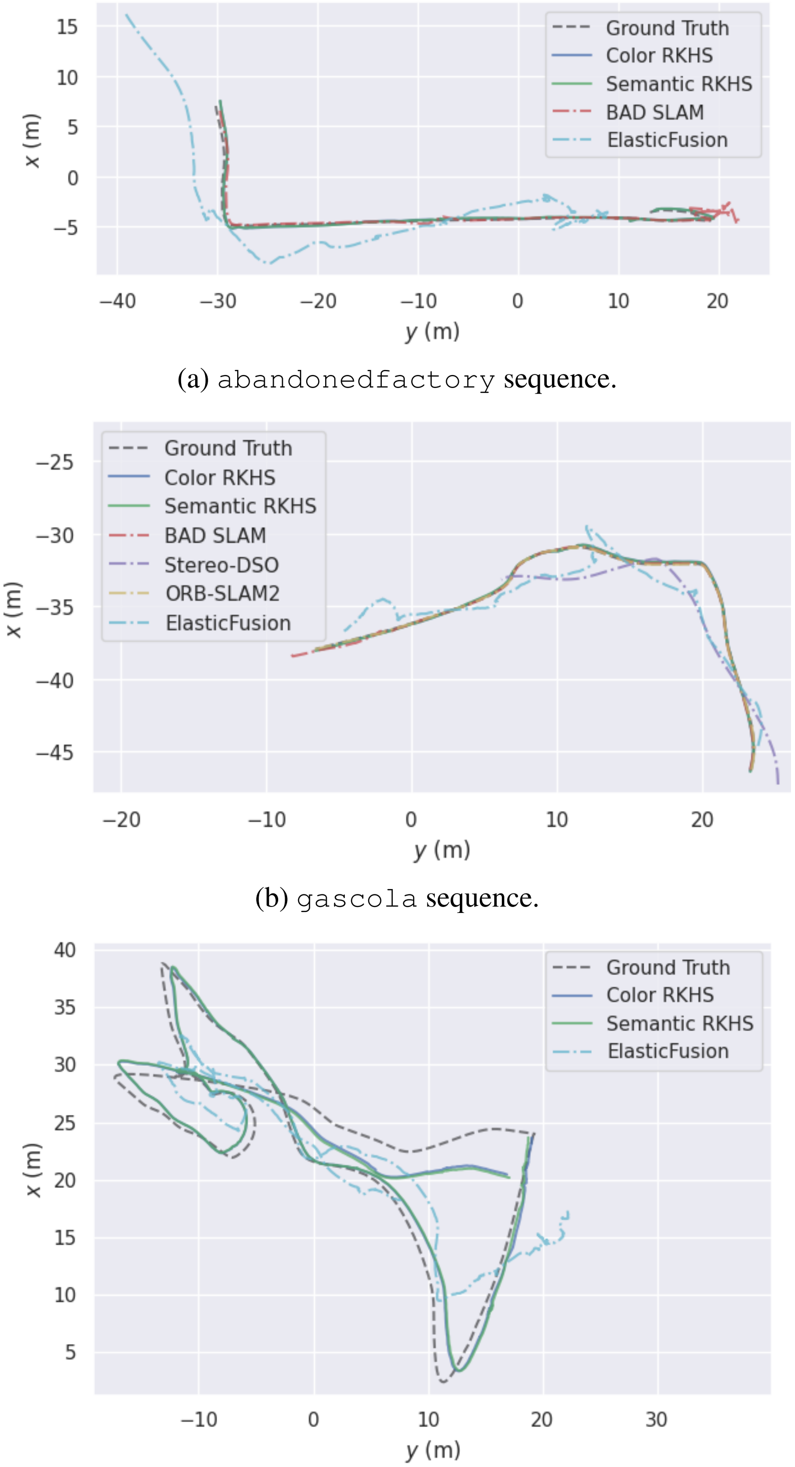
5.5.3.3 Experiment Results

The quantitative comparisons between the proposed method and the baselines are listed in Figure 29. We run the proposed methods with color and semantic features. The proposed method has lower translation errors on all the sequences but `seq 00`.

5.6 Discussions and Limitations

Besides the baseline results reported above, we also test BAD-SLAM with full loop closures. Its translation drifts in RPE are significantly reduced, with a mean RPE of $0.7240\mathrm{m}/\sec$ to $0.1273\mathrm{m}/\sec$ on `fr1` and from $0.6121\mathrm{m}/\sec$ to $0.1389\mathrm{m}/\sec$ on `fr3`, but still not as accurate as RKHS-BA. Also, we test DSO's photometric backend with the same frontend tracking as RKHS-BA on the TartanAir Dataset, but the improvement on the final ATE error on the `gascola` sequence is marginal, from $5.4988m$ to $5.4895m$, while still not able to complete other sequences. This indicates that its photometric bundle adjustment is less robust than the proposed method in highly semi-static environments.

In the experiments, we notice that the initial lengthscale choice affects the gradient calculation. The traditional energy functions have larger values when the point clouds are far away. However, if the initial lengthscale is not large enough in RKHS-BA, the proposed formulation will have smaller inner product values in the same situation, which will lead to vanishing gradients. To address this problem, the optimization starts with a sufficiently large lengthscale at the cost of more computation time.

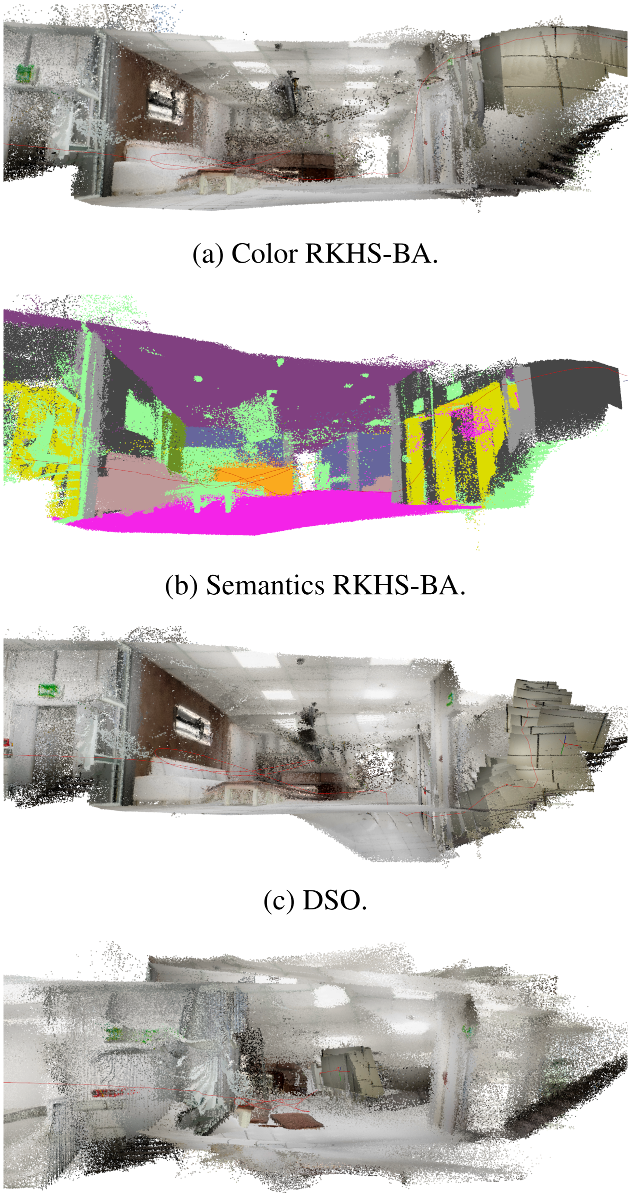


(a) `abandonedfactory` sequence.

(b) `gascola` sequence.

(c) `soulcity` sequence.

Figure 27: Trajectories of the proposed method (solid line), baselines (dash-dot line), and ground truth (dashed line) on three TartanAir [194] sequences. Only baselines that successfully complete the sequences are plotted.



(a) Color RKHS-BA.

(b) Semantics RKHS-BA.

(c) DSO.

(d) Elastic Fusion.

Figure 28: Qualitative comparisons of the stacked point cloud map of the four methods above in the TartanAir `hospital` sequence. We use the poses from their result trajectories and the raw point cloud inputs. RKHS-BA in (a) and (b) reconstruct the stairs and the wall on the right side consistently. DSO [59] in (c) fails to reconstruct the wall on the right, and the floor is cracked. ElasticFusion [202] in (d) can hardly show the structure of the hospital rooms. ORB-SLAM2 [121]'s result is not plotted because it doesn't complete the sequence.

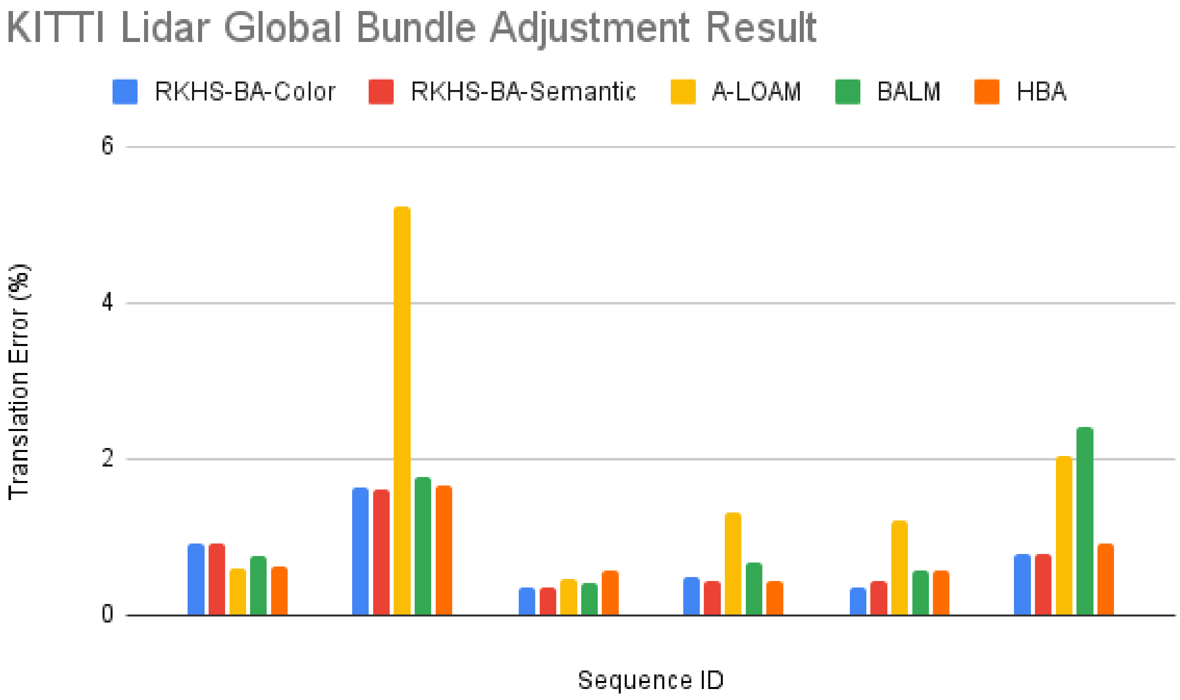


Figure 29: We compare the proposed RKHS-BA of color and semantic features with other state-of-the-art Lidar local and global bundle adjustment methods [220, 106, 105] on six KITTI [69] Lidar sequences, `00, 02, 05, 06, 07, 09`. All the methods start from the same initial trajectories and the same downsampled point clouds. The proposed method has lower translation errors on all the sequences except `seq 00`.

CHAPTER 6

EquivCVO: Correspondence-Free SE(3) Point Cloud Registration with Unsupervised Equivariance Learning

6.1 Overview

This chapter presents a novel unsupervised learning framework for correspondence-free 3D point cloud registration, leveraging SE(3)-equivariant networks to accurately recover continuous SE(3) transformations. Traditional methods, often challenged by arbitrary 3D motions and sensor noises in point cloud observations, rely primarily on invariant feature matching or global equivariant feature pooling. Our method diverges by treating point clouds as nonparametric functions in a reproducing kernel Hilbert space (RKHS), each point described by high-dimensional SE(3) steerable features. The process is structured based on the Continuous Visual Odometry (CVO) formulation, which uses 3D coordinates. The minimization in feature space circumvents the need for pairwise point correspondences. The proposed unsupervised inner-outer loop learning strategy excels in environments with limited ground truth data, offering robust adaptation in function space and enhanced resilience against outliers and mismatches. This robustness is evident in various real-world scenarios. The framework's effectiveness is demonstrated through superior performance over existing baselines on both the ModelNet simulated dataset and the ETH3D real-world RGB-D dataset.

6.2 Introduction

Point cloud registration is crucial for determining the relative transformation between two sets of 3D spatial observations, as extensively studied in literature [9, 28, 111, 223, 229]. It is commonly formulated as a nonlinear optimization problem, with data inputs from varied sensors such as RGB-D cameras, stereo cameras, and LIDAR. This technique is vital in computer vision and robotics, especially for applications in visual odometry [95] and 3D reconstruction [202]. Despite its wide use, point cloud registration encounters numerous challenges. These include complexities in nonlinear optimization on Riemannian manifolds, addressing non-overlapping observations, and mitigating the impact of sensor noise and outliers. These challenges stem from two tightly decoupled components in traditional point cloud registration: point representations and correspondences. Point representation refers to the actual format of the raw point data in feature space, while correspondences deal with optimizing the residuals from point pairings, which are influenced by the point representations.

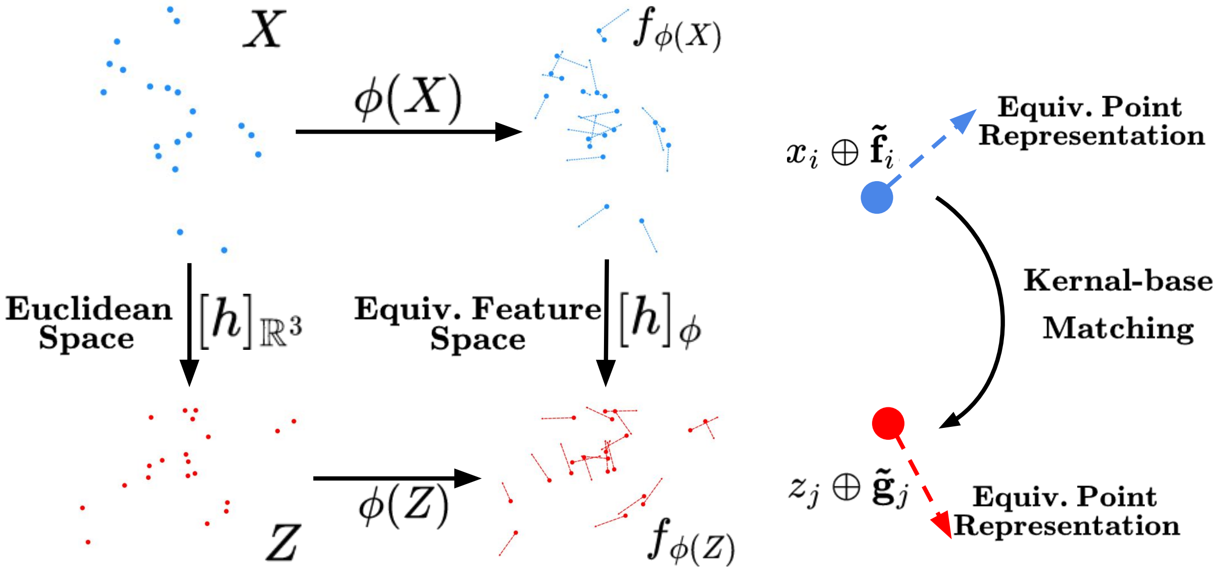


Figure 30: Registration in RKHS with Equivariant Features: The registration process takes equivariant feature embeddings $\phi(X)$ and $\phi(Z)$ from point clouds X and Z. The point cloud embeddings are further represented as continuous functions $f_{\phi(X)}$ and $f_{\phi(Z)}$ in RKHS, allowing for the utilization of a distance metric, $\|f_{\phi(X)} - f_{\phi(Z)}\|^2_{\mathcal{H}}$, for direct pose optimization in the feature space.

Classical registration methods represent points using hand-crafted geometric primitives such as 3D point coordinates [9, 159], planes [28], Gaussian mixtures [111, 91], and surfels [202, 27]. These representations, typically low-dimensional vectors, allow residuals to be computed directly as Euclidean or Mahalanobis distances. However, they often struggle with handling noise, outliers, and non-overlapping observations due to their limited expressiveness. Overcoming this limitation requires sufficient correct data correspondence estimations and robust optimization strategies to minimize these limitations [210].

In contrast, recent advancements leverage deep neural networks, capable of learning point representations that embody geometric invariance [139, 103, 196, 32]. This approach offers enhanced expressiveness for data association and improved robustness against noise and outliers in real-world scenarios [217, 86]. These methods focus on learning point-wise local and global features that remain invariant under pose transformations, facilitating semantic-aware data association. Once 1-to-1 correspondences are established, RANSAC or weighted SVD are employed for pose regression [62, 9]. However, challenges persist in generalizing invariant learning to avoid excessive data augmentation and reliance on extensive ground truth data. Additionally, efficiently correlating with transformations in $\mathrm{SE}(3)$ space and managing the computational demands of sampling transformations in the $\mathrm{SE}(3)$ space remain significant hurdles.

Equivariant features [36, 38, 198, 179] provide an alternative deep representation for point clouds. *Equivariance* is a property for a map such that given a transformation in the input, the output changes in a predictable way determined by the input transformation: A function $f : X \to Y$ is **equivariant** to a set of transformations G, if for any $g \in G$,

$$g_Y f(x) = f(g_X x), \forall x \in X \ .$$

For example, applying a translation on a 2D image and then a convolution layer is identical to processing the original image with a convolution layer and then shifting the output feature map; hence, convolution layers are natively translation-equivariant [36]. Recent strides in equivariant learning have expanded to include $\mathrm{SO}(3)$ [48, 230], $\mathrm{SE}(3)$ [23, 229], and $\mathrm{E}(n)$ [155] equivariant networks. These networks extend equivariance to more complex transformations in 3D point clouds, reducing the need for data augmentation and labeling and thereby leading to improved generalization[229] over invariant learning-based architectures. Equivariant learning has shown promise in current applications within physics, chemistry, and simulated robotics. Yet, its effectiveness in real-world point cloud registration is not well-established. For existing works, common practices include training a shape embedding to re-establish the 1-to-1 correspondence [230], or pooling point-wise equivariant features into global equivariance features [23, 229]. This approach may undermine the complexities of the noisy and outlier-rich real data where exact symmetry might not hold.

In this work, we develop an unsupervised deep equivariant feature learning and $\mathrm{SE}(3)$ registration framework (Figure 30) for point clouds, focusing on learning point-wise representations that respect the intricate geometric structure in feature space. Our approach interprets the feature embeddings of these point clouds as nonparametric functions within a reproducing kernel Hilbert space (RKHS). This unique perspective allows for feature space registration without strict correspondences and facilitates a comprehensive evaluation of its performance across various datasets. The key contributions of our work are outlined as follows:

- A novel unsupervised correspondence-free SE(3) point cloud registration framework with a lightweight SE(3) equivariant feature representation.

- An equivariant feature-based function construction methodology in an RKHS.
- An unsupervised inner-outer loop learning scheme for iterative optimization in the equivariant feature space. The inner loop performs iterative pose optimization while the outer loop iteratively updates the equivariant encoder with curriculum learning.
- Validation of the proposed contributions' effectiveness on classical and learning-based point cloud registration baselines across synthetic and real-world datasets.

6.3 Problem Formulation

6.3.1 Problem Definition and Notations

Consider two (finite) collections of points, $X = \{x_i\}$, $Z = \{z_j\} \subset \mathbb{R}^3$. We aim at finding an element $h \in \mathrm{SE}(3)$, which minimize a distance metric between two point clouds X and $hZ = \{hz_j\}$:

$$\hat{h} = \arg \min_{h \in \mathrm{SE}(3)} d(X, hZ) \, . \tag{6.1}$$

6.3.2 Registeration in Reproducible Kernel Hilbert Space

In preparation for delving into the details of our *EquivCVO* in the following section, it is beneficial to briefly review the notations and core principles outlined in the CVO framework [34, 223].

The point clouds, X and Z, are first represented as functions $f_X, f_Z : \mathbb{R}^3 \to \mathcal{I}$ that live in some reproducing kernel Hilbert space (RKHS), $(\mathcal{H}, \langle \cdot, \cdot \rangle_{\mathcal{H}})$. The group action $\mathrm{SE}(3) \curvearrowright \mathbb{R}^3$ induces an action on the RKHS, $\mathrm{SE}(3) \curvearrowright \mathcal{H}$, denoted as $h.f(x) := f(hx)$. Inspired by this observation, we will set $h.f_Z := f_{hZ}$. Furthermore, each point might contain pose-invariant information in different dimensions, such as color or intensity, described by a point in an inner product space, $(\mathcal{I}, \langle \cdot, \cdot \rangle_{\mathcal{I}})$. We introduce two labeling functions, $\ell_X : X \to \mathcal{I}$ and $\ell_Z : Z \to \mathcal{I}$ for the two point clouds, respectively. With the kernel trick [13], the point cloud functions are

$$\begin{aligned} f_X(\cdot) &:= \sum_{x_i \in X} \ell_X(x_i) k_\ell(\cdot, x_i), \\ f_{hZ}(\cdot) &:= \sum_{z_j \in Z} \ell_Z(z_j) k_\ell(\cdot, hz_j). \end{aligned} \tag{6.2}$$

where the kernels are symmetric and positive definite functions parameterized by some parameter ℓ: $k_\ell : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$.

To measure the alignment of the two point clouds, we use the distance between two point cloud functions

$$\begin{aligned} d(f_X, f_{hZ}) &= \|f_X - f_{hZ}\|^2_{\mathcal{H}} \\ &= \langle f_X, f_X \rangle + \langle f_Z, f_Z \rangle - 2\langle f_X, f_{hZ} \rangle. \end{aligned} \tag{6.3}$$

The distance is well-defined because RKHS is endowed with a valid inner product. With the *reproducing property* [8], each inner product can be further broken into

$$\langle f_X, f_{hZ} \rangle_{\mathcal{H}} = \sum_{x_i \in X, z_j \in Z} \langle \ell_X(x_i), \ell_Z(z_j) \rangle k_\ell(x_i, hz_j) \ .$$

6.4 EquivCVO

Figure 30 illustrates the *EquivCVO* framework. The process begins with introducing a lightweight SE(3) equivariant feature representation as detailed in Section 6.4.1. Subsequently, we focus on optimizing the pose and kernel parameters within this feature space (Sections 6.4.2 and 6.4.3). The training phase is distinctive due to the disparate stages and frequencies at which the equivariant feature encoder and the kernel and pose updates occur. To address this, we perform an unsupervised inner-outer loop learning strategy with curriculum learning, as discussed in Section 6.4.4.

6.4.1 Equivariant Point Representation

Unlike raw 3D coordinates, feature maps extracted from deep neural networks produce a more expressive representation of the point clouds. Instead of representing each point as an element in $\mathbb{R}^3$ as in CVO, we choose equivariant features to represent them, $x \oplus \tilde{\mathbf{f}}$: a direct sum of x's coordinate and multiple channels of 3-dimensional steerable vectors $\tilde{\mathbf{f}} := \phi(x)$, while ϕ being the equivariant encoder with weights θ. The steerable features are a specific type-1 feature [179] for rotations in VectorNeuron [48]. VectorNeuron proposes that $\mathrm{SE}(3)$ equivariance can be realized by centering the point cloud coordinates. However, in real applications, the two input point clouds do not fully overlap. Thus, we cannot simply centralize them and process the rotation-only registration. Instead, we add an additional type-0 feature, the 3D coordinate itself. Each channel of this feature representation can be visualized as a vector field defined on $\mathbb{R}^3$, as demonstrated in Figure 31.

We can apply the rotation R and translation t of the pose h directly on the point-wise feature

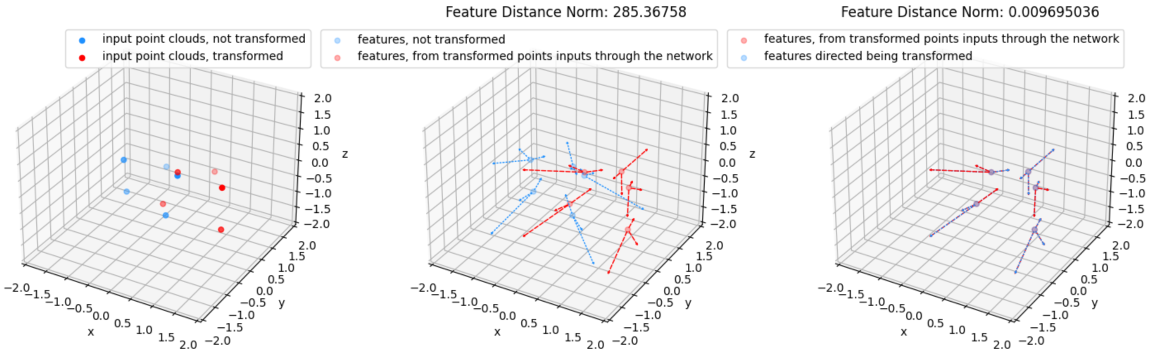


Figure 31: Equivariant Representation of Point Feature: (Left) Visualization of the two raw input point clouds in blue and red, being the 3D coordinate itself; (Middle) The direct sum representation of equivariant point features of the two point clouds at the initial relative pose, where each point attaches its steerable vectors; (Right) Post ground truth SE(3) transformation applied directly to the feature space, resulting in an exact overlap of the two representations of the point set, affirming the precision of the equivariant representation.

representations as follows:

$$R(x \oplus \tilde{\mathbf{f}}) = Rx \oplus R\tilde{\mathbf{f}}, \tag{6.4}$$

$$t(x \oplus \tilde{\mathbf{f}}) = (t + x) \oplus \tilde{\mathbf{f}}. \tag{6.5}$$

The rotation's action on the vector field is on both the coordinates and the steerable features. The translation's action will only alter the coordinates but will not affect the vector field elements' directions.

We can define the linear multiplications by weights W as well as the nonlinearity that only acts on the steerable features. We use the same non-linearity as in VectorNeuron [48]

$$W(x \oplus \tilde{\mathbf{f}}) = x \oplus (\tilde{\mathbf{f}}W), \tag{6.6}$$

$$\sigma(x \oplus \tilde{\mathbf{f}}) = x \oplus \sigma(\tilde{\mathbf{f}}), \tag{6.7}$$

We have the following convolution

$$W * (x \oplus \tilde{\mathbf{f}}) = x \oplus \sigma(\sum \tilde{\mathbf{f}} W_i). \tag{6.8}$$

The neighboring points include those whose steerable features are similar to the current point in the feature space.

6.4.2 Pose Optimization and Kernel Learning in the Feature Space

To estimate the pose, we want to minimize the distance of the two functions in the RKHS:

$$\begin{aligned} d(f_{\phi(X)}, f_{h\phi(Z)}) =& \|f_{\phi(X)}\|^2 + \|f_{\phi(Z)}\|^2 \\ & - 2\langle f_{\phi(X)}, f_{h\phi(Z)}\rangle, \end{aligned} \tag{6.9}$$

where each function is represented as $f_{\phi(X)} = \sum l_X(x_i) k_\ell(x_i \oplus \tilde{\mathbf{f}}_i, \cdot)$. We denote $\tilde{\mathbf{f}}_i := \phi(x_i)$ and $\tilde{\mathbf{g}}_j := \phi(hz_j) = h\phi(z_j)$. Then we have

$$\begin{aligned} & d(f_{\phi(X)}, f_{h\phi(Z)}) = \\ & \sum_{i,j} \langle (l_X(x_i), l_X(x_j))\rangle k_\ell(x_i \oplus \tilde{\mathbf{f}}_i, x_j \oplus \tilde{\mathbf{g}}_j) \\ & + \sum_{i,j} \langle l_Z(z_i), l_Z(z_j))\rangle k_\ell(z_i \oplus \tilde{\mathbf{g}}_i, z_j \oplus \tilde{\mathbf{g}}_j) \\ & - 2\sum_{i,j} \langle l_X(z_i), l_Z(z_j))\rangle k_\ell(x_i \oplus \tilde{\mathbf{f}}_i, z_j \oplus \tilde{\mathbf{g}}_j)\ . \end{aligned} \tag{6.10}$$

As the label function (color, intensity, etc) is invariant to the poses, we can consider them as constants, $c_{ij}^X, c_{ij}^Z, c_{ij}$:

$$\begin{aligned} d(f_{\phi(X)}, f_{h\phi(Z)}) =& \sum_{i,j} c_{ij}^X k_\ell(x_i \oplus \tilde{\mathbf{f}}_i, x_j \oplus \tilde{\mathbf{f}}_j) \\ & + \sum_{i,j} c_{ij}^Z k_\ell(z_i \oplus \tilde{\mathbf{g}}_i, z_j \oplus \tilde{\mathbf{g}}_j) \\ & -2\sum_{i,j} c_{ij} k_\ell(x_i \oplus \tilde{\mathbf{f}}_i, z_j \oplus \tilde{\mathbf{g}}_j)\ . \end{aligned} \tag{6.11}$$

During the inference stage, we want to minimize both the distance between two functions with respect to the pose h as well as kernel parameter ℓ, while holding the encoder weights θ fixed:

$$\hat{h}, \hat{\ell} = \arg\min_{h,l} d(f_{\phi(X)}, f_{h\phi(Z)}) \tag{6.12}$$

Note that for each iteration of the pose optimization, we don't need to resend the transformed point cloud through the encoder again. Instead, we just need to directly transform the equivariant features and re-evaluate the kernels in the loss.

6.4.3 Kernel Choice

The RKHS that the point cloud functions living in requires a properly defined Mercer kernel [8], which is a function of two variables in the equivariant feature space

$$k_\ell : \phi \times \phi \to \mathcal{H} \tag{6.13}$$

It is symmetric and positive-definite. We define our kernel as the product of the RBF kernel and the hyperbolic tangent kernel [143]:

$$k_\ell(x_i \oplus \tilde{\mathbf{f}}_i, z_j \oplus \tilde{\mathbf{g}}_j) := \mathrm{RBF}_\ell(x_i, z_j) \cdot \tanh\left(1 + \tilde{\mathbf{f}}_i \cdot \tilde{\mathbf{g}}_j\right) \tag{6.14}$$

because the product of two kernels is still a kernel. We use the RBF kernel for the coordinate part and the hyperbolic tangent kernel for the steerable feature maps. The RBF kernel has a kernel parameter, the lengthscale ℓ, to be optimized during the pose inference:

$$\mathrm{RBF}_\ell(x_i, z_j) = \exp(\frac{\|x_i - z_j\|_3^2}{2\ell^2}), \tag{6.15}$$

while the hyperbolic tangent kernel does not. We adopt the RBF kernel to use the lengthscale parameter to encourage sparsity and reduce the number of non-trivial terms in the loss. We do not choose a parameterized kernel for steerable vectors $\tilde{\mathbf{f}}$ because we want to reduce the number of parameters to optimize during test time.

6.4.4 Unsupervised Training of Equivariant Encoder

In real applications like visual odometry, limited ground truth pose labels are available. To adapt the encoder weights to new environments, we choose to perform unsupervised bi-level training [135].

$$\text{Inner Loop} : \arg\min_{h,l} d(f_{\phi(X)}, f_{h\phi(Z)}) \tag{6.16}$$

$$\text{Outer Loop} : \arg\min_{\theta} d(f_{\phi(X)}, f_{\hat{h}\phi(Z)}) \tag{6.17}$$

In training, we first send the two point clouds X, Z through the equivariant encoder ϕ to obtain the point-wise equivariant features $\phi(X), \phi(Z)$. Then, in each iteration, we minimize the loss with respect to the pose h and kernel parameter ℓ to produce a step pose update. Based on the latest pose estimate $\hat{h}$, we keep the gradient in the computation graph and update the encoder parameters. This training strategy doesn't require the ground truth pose label.

We have some further considerations of the training procedure. We use the curriculum training strategy to bootstrap the training when we start from random initial weights. We start from smaller angles at 1° and gradually towards larger angles at 90°. Besides, the kernel parameter will occasionally change too fast and effectively become all zero. To prevent this from happening, we use a 100 times smaller learning rate for updating the kernel lengthscale ℓ.

6.5 Experiments

In this section, we present qualitative and quantitative experimental results on the simulated ModelNet dataset [207] and real ETH3D RGB-D dataset [158]. We evaluate *EquivCVO* 's registration accuracy in rotations and translations, robustness, and generalization capability. For each dataset, we use the same set of hyperparameters.

Baselines: We choose three types of baselines. a) Classical non-learning registration methods, including ICP [9] and GICP [159].b) Invariant feature-matching based methods, including RANSAC [62] with FPFH features and FGR [226] with FPFH features. c) Equivaraint finite-group-based feature pooling method, E2PN [229], trained under the same setup as *EquivCVO* .

6.5.1 Simulation Dataset: ModelNet Registration

Setup: In this toy example, we perform point cloud registration of *EquivCVO* on the ModelNet40 dataset. It contains shapes generated from 3D CAD models. To avoid the pose ambiguity of objects with symmetric rotational shapes, only the airplane category is used in this experiment, with 60% training data, 20% validation data, and 20% test data. A point cloud is generated by randomly subsampling 1,024 points on the surface and it is randomly rotated to form a pair. We set the initial rotation angle at 45° and 90° around random axes. To study the model generalization under different types of noise perturbations during inference time, we inject three types of noises: a) Gaussian noise $\mathcal{N}(0, 0.01)$ distributed along each point's surface normal. b) 20% uniformly distributed outliers along each point's surface normal c) 10% random cropping along a random axis. These noises are *not* applied during training time.

Results: The quantitative results are presented in Table 6.1 and the qualitative results are shown in Figure 32. We denote the variance of the Gaussian noise as σ and the ratio for the uniform outlier perturbation as γ.

In noise-free conditions, both classical and proposed methods excel at smaller angles (45°), with invariant feature-matching methods showing lower errors than equivariant-learning-based approaches. *EquivCVO* demonstrates performance on par with classical ICP methods and superior to E2PN. However, at initial angles of 90°, ICP and GICP exhibit larger errors due to their reliance on accurate initial guesses for data association. In these scenarios, *EquivCVO* surpasses E2PN,

Type	Method	Test Init Angle $< 45^\circ$			Test Init Angle $< 90^\circ$		
		$\sigma = 0, \gamma = 0$	$\sigma = 0.01, \gamma = 0$	$\sigma = 0.01, \gamma = 20\%$	$\sigma = 0, \gamma = 0$	$\sigma = 0.01, \gamma = 0$	$\sigma = 0.01, \gamma = 20\%$
Non-Learning	ICP	0.25	0.37	0.62	34.52	35.72	39.69
	GICP	0.49	0.45	0.74	42.33	43.64	37.82
Invariant Features	FPFH + RANSAC	0.00	42.20	42.45	0.00	85.40	85.34
	FPFH + FGR	0.04	1.22	18.72	0.14	4.83	41.45
Equivariant Features	E2PN	1.46	3.70	6.75	1.42	3.17	7.59
	EquivCVO	0.29	1.12	4.02	1.07	2.70	4.77

Type	Method	Test Init Angle $< 45^\circ$			Test Init Angle $< 90^\circ$		
		crop 5%	crop 10%	crop 20%	crop 5%	crop 10%	crop 20%
Non-Learning	ICP	1.21	1.84	4.61	38.04	41.25	42.00
	GICP	1.17	2.51	1.26	32.96	37.14	41.27
Invariant Features	FPFH + RANSAC	42.83	42.75	43.10	85.89	86.17	85.74
	FPFH + FGR	53.88	57.69	77.31	67.42	80.10	90.87
Equivariant Features	E2PN	14.80	22.31	18.76	24.19	18.05	21.80
	EquivCVO	2.69	7.07	14.48	8.76	14.80	30.10

Table 6.1: Rotation Error Analysis on the ModelNet Dataset: (Top) Comparative performance of baselines under varying noise and outlier conditions. σ is the variance of the Gaussian noise applied on the surface normal direction of each point. γ is the ratio of points perturbed by uniformly distributed outliers. (Bottom) Baseline comparison across different crop ratios.

though invariant feature-matching methods achieve the most favorable outcomes.

When encountering Gaussian noise, *EquivCVO* reaches a slightly better accuracy than the invariant feature matching methods at 45° and is the best-performing method at 90°. ICP-based methods still top the benchmark at 45°, but similar to the noise-less situation, its result degenerates at larger initial angles. RANSAC is not doing well in this situation.

With the introduction of 20% uniformly distributed outliers, methods that assume Gaussian errors will degenerate. Invariant feature matching is severely affected by this type of perturbation and fails to register at smaller or larger angles. ICP-based methods can reach satisfactory results at small angles but not larger ones. Equivariant learning-based methods are not heavily affected by this perturbation, while *EquivCVO* has an edge over the other equivariant baseline. This demonstrates how the expressiveness of equivariant features helps in the robustness of the registration process, even when only noise-free data is used in training.

In tests involving random cropping of input data (with no cropping in training), as reported in Table 6.1 (Bottom), all methods experience performance dips. Similar to the third case, ICP-based methods are not substantially affected by the cropping at 45° but are easily trapped in the local minima at larger angles. Invariant feature-based baselines cannot converge at either initial angle. Both equivariant methods also experienced larger errors, though not as severe as the invariant feature matching methods. The proposed learning-based RKHS formulation natively annihilates the outlier disturbance because, at larger distances, the kernel will return trivial values. In contrast, as E2PN directly performs global pooling over all the points to obtain a single global feature,

Type	Method	Rotation Error $(^\circ)$		Translation Error (m)	
		Mean	STD.	Mean	STD.
Non-Learning	ICP	0.88	1.30	0.03	0.05
	GICP	0.69	3.54	0.02	0.11
InvariantFeatures	FPFH + RANSAC	8.75	2.95	0.17	0.40
	FPFH + FGR	3.60	12.61	0.08	0.17
Equivariant Features	E2PN	5.20	0.00	NA	NA
	EquivCVO	0.55	1.22	0.02	0.05

Table 6.2: Rotation and Translation Error Results on the ETH3D Dataset: Compared with other baseline methods, *EquivCVO* achieves the best rotation and translation errors. E2PN is $\mathrm{SE}(3)$ equivariant, but the implementation we use does not yet have translation predictions for comparison.

missing cropped components will reduce the quality of the global feature, especially when the crop is unseen in the training data.

6.5.2 Real Dataset: ETH3D RGB-D Registration

Setup: In this experiment, we benchmark *EquivCVO* in a real RGB-D dataset. We use the ETH3D dataset [158], which contains real indoor and outdoor RGB-D images. In this setup, two point cloud pairs are sampled sequentially. Unlike the simulated ModelNet dataset, a pair of point clouds will not fully overlap even without noise injections due to the viewpoint change. Additionally, the ground truth pose will contain rotation and translation but at smaller angles than the ModelNet experiment. We use 6 sequences for training: (`cable_3`, `ceiling_1`, `repetitive`, `einstein_2`, `sfm_house_loop`, `desk_3`), 2 sequences for validation (`mannequin_3`, `sfm_garden`), and 4 sequences for testing (`sfm_lab_room_1`, `plant_1`, `sfm_bench`, `table_3`). As the number of frames in each sequence is not uniform, we subsample the frame pairs such that 1000 pairs are selected per sequence at most. This results in 5919 training instances, 2000 validation instances, and 2702 test instances. For all the methods, we downsample the input point clouds into 1024 points with the `farthest_point_down_sample()` from Open3D for all methods. For the fair comparison, we do not use the color information by setting the label function $l_X(x) = 1$ in Eq. (6.2) as the baselines do not use color either.

Results: The quantitative results are shown in Table 6.2. On the test sequences, *EquivCVO* demonstrates the best accuracy in both rotation and translation evaluations, with a 0.55° rotation error and $0.02m$ translation error. The invariant feature-based baselines have significantly larger test errors. ICP-based methods have comparable translation errors, but their rotation error is 60% and 25% larger, respectively. This comparison indicates that *EquivCVO* produces fine-grained registration alone in real data and thus can be directly adopted in applications like point cloud pose tracking. It does not have the necessity of using coarse-to-fine strategies with ICP, as adopted in recent invariant-learning-based works like PREDATOR [86].

Moreover, the other equivariant baseline, E2PN, is also not as accurate as *EquivCVO* , though it is also correspondence-free. We argue that there are three potential reasons behind this: First, E2PN chooses the finite group rotation representation on equivariance learning, resulting in a much faster running speed via feature permutation. However, the discretization comes at a cost; that is, it will have resolution challenges at fine-grained registration, especially compared to *EquivCVO* 's continuous rotation representation. Secondly, *EquivCVO* does not require training labels and thus is not tightly coupled to the training data distribution. In contrast, E2PN needs ground truth supervision, which means there would be overfitting challenges if the test set is a new scene. Thirdly, *EquivCVO* adopts the RKHS representation whose kernel can eliminate the influence of non-overlapped areas, while E2PN assumes complete symmetry of the input pair, which is often violated in real data. Recent works such as SE3-Transformer [65] and GeoTransformer [141] attempt to bring the attention mechanism to address this issue. But training the attention network will also need ground truth labels.

6.5.3 Ablation Study

6.5.3.1 Comparisons with Classical CVO

As shown in Table 6.3, we compare *EquivCVO* with CVO [34, 223], which shares the same correspondence-free RKHS loss but does not learn equivariant features. The original CVO has demonstrated superior robustness; therefore, we perform the comparison with the existence of Gaussian noise and 20% uniformly distributed outliers on the ModelNet dataset. The result indicates that the unsupervised kernel learning of the equivariant features has effectively improved the registration accuracy and reduced the uncertainty. However, the original CVO implementation is significantly faster per iteration because it does not need to evaluate a high-dimensional kernel computation of steerable features.

6.5.3.2 Initial Kernel Parameter

EquivCVO has a hyperparameter, the kernel lengthscale ℓ, which controls the coarse-grain and fine-grain resolution of the loss [34, 223]. It is optimized during pose regression but still requires an initial value. In this ablation study shown in Table 6.4, we test how the init lengthscale will affect the registration accuracy on the ModelNet dataset. The result aligns with the finding from the CVO technical reports: When registering at a larger angle, a larger initial lengthscale is needed for a global perspective.

Method	Init Rotation: 45°		Init Rotation: 90°		Time per Iteration $(\mathtt{s})$
	Mean	STD.	Mean	STD.	
EquivCVO	2.55	5.75	3.20	28.90	$0.6\mathtt{s}$
CVO	19.73	24.24	23.65	34.40	$0.01\mathtt{s}$

Table 6.3: Comparisons between *EquivCVO* and Classical CVO: On the ModelNet dataset, *EquivCVO* demonstrates significantly smaller mean and STD for both rotation and translation. However, it is important to note that the unoptimized implementation of *EquivCVO* exhibits considerably slower computational performance than CVO.

Init lengthscale (ℓ)	0.25	0.5	0.75	1.0
Init Angle: 45°	48.2	1.12	0.93	0.29
Init Angle: 90°	68.01	8.77	5.95	1.07

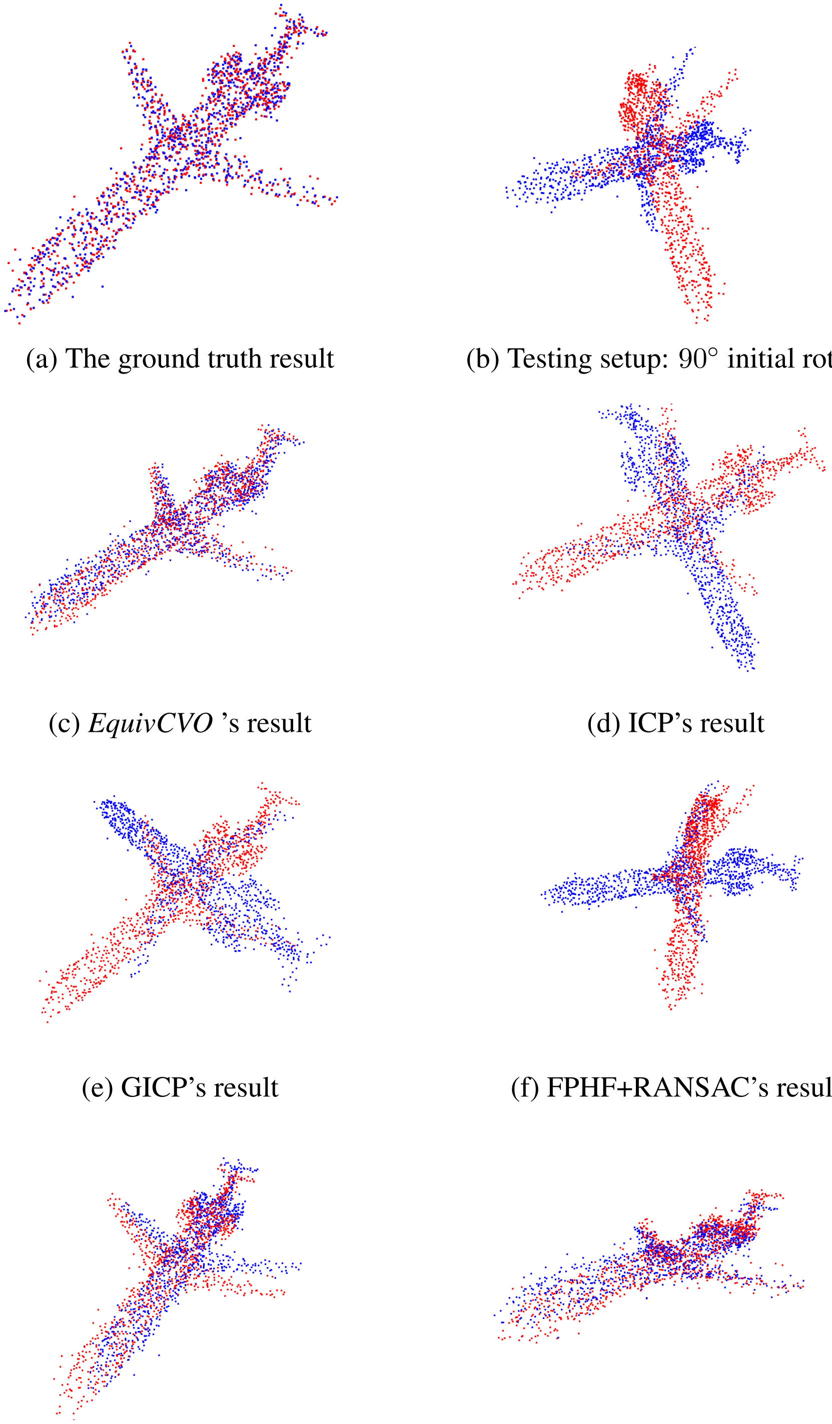
Table 6.4: Ablation Study on Kernel Lengthscale: This study examines the effects of four distinct kernel lengthscales on two initial angles. The findings suggest a positive correlation between the problem scale and the kernel lengthscale, indicating that larger initial errors necessitate a correspondingly larger lengthscale.

6.5.3.3 Is Curriculum Learning Necessary?

Being an unsupervised method, a challenge we have encountered is how to bootstrap the randomly-initialized network weights. We compare the training result between the curriculum training and the direct training at the maximum angle in Table 6.5. The result shows that using a small step size in the curriculum leads to higher accuracy and lower uncertainty.

Curriculum	$[45°]$		$[1°, 10°, 20°, 30°, 45°]$	
	Mean	STD.	Mean	STD.
Init Angle: $45°$	2.73	9.1	0.29	0.469

Table 6.5: Ablation Study on the Necessity of Curriculum Learning: The results demonstrate that initializing the *EquivCVO* on the ModelNet dataset with small, incremental angles leads to better error means and STD compared to learning directly from larger angles, highlighting the effectiveness of a curriculum learning in *EquivCVO* training.



(a) The ground truth result

(b) Testing setup: 90° initial rotation

(c) *EquivCVO* 's result

(d) ICP's result

(e) GICP's result

(f) FPHF+RANSAC's result

(g) FPHF+FGR's result

(h) E2PN's result

Figure 32: An example of the point cloud registration at 90° initial angle, with Gaussian noise $\mathcal{N}(0, 0.01)$ along the surface normal direction and 20% uniformly distributed outliers. The equivariant registrations outperform the invariant ones and ICP-based methods. *EquivCVO* has a slighly better yaw angle comparing to E2PN.

CHAPTER 7

Conclusion

This thesis considers the advancement of resilient and semantic-aware robot perception systems. We focus on constructing symmetric and spatial-semantic representations of unknown environments, as well as developing robust localization methods on top of these representations. In practical applications, challenges such as inadequate training data encompassing all potential inputs, the scale of 3D observations, sensors perturbed with noise and outliers, highly nonlinear optimizations, and unreliable semantic inputs from preceding machine learning methods are prevalent. In light of these constraints, this thesis presents several notable contributions:

- Chapter 3 discusses a semantic map compression method using sparse Bayesian regressions, particularly with the Relevance Vector Machine. We introduce a sparse spatial-semantic representation for robot mapping. It demonstrates a highly memory-efficient strategy of storing both semantic and geometric information and has comparable semantic query accuracy with mainstream mapping techniques.

- Chapter 4 introduces Semantic Continuous Visual Odometry (CVO), a nonparametric data association-free and global point cloud registration framework with reproducing kernel Hilbert space (RKHS). The framework shifts focus from raw 3D observations, instead tightly combining both geometric and invariant semantic data as a unified input representation. It also offers a new alignment metric that evaluates the compatibility of both geometric and semantic characteristics. The uniquely designed spatial-semantic alignment metric provides the ability for global rotation regression through alignment tests within the discretized rotation group. Its experimental evaluations exhibit remarkable exceptional accuracy and resilience against noisy and occluded inputs, irrespective of the size of the rotation angle.

- Chapter 5 extends the two-frame registration system in Chapter 4 to a multi-frame Bundle Adjustment (BA) formulation, the RKHS-BA. It reformulates the original nonlinear loss function, the weighted sums of exponentials, into the convex Iterative Reweighted Least Squares (IRLS) problems, thereby facilitating largescale optimization. Furthermore, we tackle the concerns of the weight explosion endemic in classical IRLS by generating the weights from spatial-semantic kernel functions that have defined bounds. Its applications have been successfully demonstrated in sliding-window visual odometry and largescale global Lidar mapping. It exhibits heightened robustness while dealing with challenging datasets varying in weather conditions, noise levels, dynamic changes, and outlier ratios.

- Chapter 6 generalizes Chapter 4 into an unsupervised deep learning framework that utilizes more expressive *equivariant* features. Observing that raw 3D points can be rotated or translated in classical registration algorithms, we develop a neural network architecture that allows similar rotation and translation actions directly on neural features of point clouds. Neural networks supporting this property are considered equivariant. With this property, the iterative registration process can directly take the highly expressive features instead of raw 3D coordinates as inputs. Notably, the network employs an unsupervised training approach, which proves beneficial in settings where ground-truth data is limited. Our benchmarks against other classical and learning-based baselines with both simulated and real-world datasets show that it surpasses them in terms of accuracy and robustness.

7.1 Limitations and Future Directions

7.1.1 Limitations

In this thesis, we have shown that our spatial-semantic point cloud registration frameworks have significant advantages over traditional localization methods in semantic awareness and robustness. We also point out what we believe are the current limitations of this framework:

- *The formulation in the BA has varying data associations at each iteration, which might lead to a long optimization time for largescale problems.* Classical BA formulation constructs least square problems from a fixed 1-to-1 data association, supporting various matrix factorization techniques in the backend that exploit the sparsity patterns. In contrast, the association in our formulation is dynamic and changing at each iteration. This trade-off has limited the running speed of the proposed methods when we are performing global bundle adjustments of thousands of frames.

- *Low-overlapped registration.* In online perception systems, a registration process can take place between a pre-existing model and the present observation, while the model's density and coverage often display substantial differences compared to the latest inputs. As shown in Chapter 4, when the overlap ratios decrease below 50%, non-learning-based methods often struggle. Recently, a class of *attention*-based [189] deep learning methods utilizes supervised learning to label the matched inliers, which means the overlapped areas are explicitly learned for the network [141, 217]. This approach has demonstrated significant improvements in the datasets featuring lower overlaps, such as the 3DMatch [219] and 3dLowMatch [86] datasets. An intriguing aspect to explore is whether it is feasible to learn spatial attention in an unsupervised manner as discussed in Chapter 6?

- *Extending the pose-only bundle adjustments to joint optimizations of maps and poses.* Tightly-coupled pose and map optimization have demonstrated sophisticated real data performance in recent localization and SLAM systems [120, 59, 202, 177]. Our current design in Chapter 5 performs the sliding window odometry with separated steps of pose regression and map fusion. Instead, the mutual data consistency for map elements across covisible frames can be integrated into the bundle adjustment formulation as well.

7.1.2 More Expressive and Generalized Equivaraint Features

The work of Chapter 6 on the registration with deep SE(3) equivariant features uncovers some interesting potential directions for future research. Point cloud registrations, which have applications in loop closures and robot state initialization, often involve non-trivial scale differences between input pairs. Although the semantic features could aid in the scale regression, one pathway to explore is that the deep networks can capture the global feature scales, in other words, equivariant to $\mathrm{Sim}(3)$ actions. Furthermore, in the context of visual odometry applications, modern depth sensors such as RGB-D cameras remain noisy and have limited ranges. Outdoor RGB-D SLAM systems sometimes opt for image-based odometry [20, 193, 177] instead of point cloud-based odometry methods, while relying on map point triangulation [120] to reconstruct the depth values. This prompts the discussion of running monocular odometry in the equivariant feature space as well, with $\mathrm{SL}(3)$ equivariant networks.

7.1.3 Robust Localization with Implicit Map Representations

Recent advances in implicit geometric models offer an alternative way of depicting spatial structures. Techniques such as NeRf [197] and Gaussian Splatting [94] have the potential to recreate more intricate details and complex illuminations compared to classical voxel-based maps, assuming sufficient observations of the target objects and known camera positions. Some early progress of implicit mapping in SLAM systems [169, 146] demonstrates the feasibility of pose estimations while constructing photo-realistic maps in indoor environments. This naturally leads to the following question: Can we execute robust global registrations of implicit map representations as well in a complete SLAM system?

APPENDIX A

Approximating the RKHS Multi-frame Registration with IRLS

In this section, we provide detailed derivations on how we approximate the full objective function in (5.7)

$$\begin{aligned} F(\mathcal{T}) &:= \sum_{(m,n)\in\mathcal{C}} \sum_{\mathbf{x}_i^m\in X_m, \mathbf{z}_j^n\in Z_n} \underbrace{k(\mathbf{T}_m\mathbf{x}_i^m, \mathbf{T}_n\mathbf{z}_j^n)\cdot c_{ij}^{mn}}_{F_{ij}^{mn}} \\ \mathcal{T}^* &= \arg\max_{\mathcal{T}}\ F(\mathcal{T}). \end{aligned} \tag{A.1}$$

with the Iterative Reweighted Least Square problem (IRLS) and solve it with the Gauss-Newton method.

With a good initialization of the frame poses $\mathcal{T} = \{\mathbf{T}_1, ..., \mathbf{T}_K\}$ from tracking, and let $d(\mathbf{x}, \mathbf{z}) := \mathbf{x} - \mathbf{z}$, we can expand each term F_{ij}^{mn} as follows:

$$\begin{aligned} F_{ij}^{mn} &= k(\mathbf{T}_m\mathbf{x}_i^m, \mathbf{T}_n\mathbf{z}_j^n)\cdot c_{ij}^{mn} && \text{(A.2)} \\ &= c_{ij}^{mn}\sigma^2 \exp\left(\frac{-\|\mathbf{T}_m\mathbf{x}_i^m - \mathbf{T}_n\mathbf{z}_j^n\|_3^2}{2\ell^2}\right) && \text{(A.3)} \\ &:= c_{ij}^{mn} k(d(\mathbf{T}_m\mathbf{x}_i^m, \mathbf{T}_n\mathbf{z}_j^n)^2) && \text{(A.4)} \end{aligned}$$

If we apply a perturbation $\boldsymbol{\epsilon}_m \in \mathbb{R}^6$ on the right of $\mathbf{T}_m$

$$\mathbf{T}_m^\star = \mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge) = \mathbf{T}_m \exp(\begin{bmatrix}\rho_m \\ \phi_m\end{bmatrix}^\wedge) \tag{A.5}$$

and then the gradient with respect to $\boldsymbol{\epsilon}_m$ is

$$\nabla F_{ij}^{mn} = c_{ij}^{mn} \frac{\partial k(d(\mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge)\mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2)}{\partial d} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} \tag{A.6}$$

$$= c_{ij}^{mn} \frac{\partial k(d(\mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge)\mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2)}{\partial d} \frac{1}{d} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} d \tag{A.7}$$

$$= c_{ij}^{mn} k \frac{-2d}{2\ell^2} \frac{1}{d} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} d \tag{A.8}$$

$$= \frac{-1}{\ell^2} \underbrace{c_{ij}^{mn} k}_{w_{ij}^{mn}} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} d \tag{A.9}$$

where we denote $w_{ij}^{mn} := c_{ij}^{mn} k(d(\mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge)\mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2)$. After summing it up for all pairs of $(m,n) \in \mathcal{C}$ and $\mathbf{x}_i^m \in X_m, \mathbf{z}_j^n \in X_n$ and taking the gradients to zero, we obtain

$$\sum_{(m,n)\in\mathcal{C}} \sum_{\substack{\mathbf{x}_i^m \in X_m \\ \mathbf{z}_j^n \in X_n}} w_{ij}^{mn} \frac{\partial d}{\partial \boldsymbol{\epsilon}_m} d = 0 \tag{A.10}$$

Here, the weight w_{ij}^{mn} encodes both the geometric and hierarchical semantic relations between the pair of points. If we treat w_{ij}^{mn} as *constant* weights during one optimization step, the solution to (A.10) corresponds to the solution for the following least square problem:

$$\arg\max_{\mathcal{T}} \sum_{(m,n)\in\mathcal{C}} \sum_{\substack{\mathbf{x}_i^m \in X_m \\ \mathbf{z}_j^n \in X_n}} w_{ij}^{mn} d(\mathbf{T}_m \mathbf{x}_i^m, \mathbf{T}_n \mathbf{z}_j^n)^2 \tag{A.11}$$

where $h \in H$ are the poses of all the keyframes involved except the first frame. To see that, we can apply the perturbation $\exp(\boldsymbol{\epsilon}_m^\wedge)$ on the right of $\mathbf{T}_m$ and then take the gradient with respect to $\boldsymbol{\epsilon}_m$ for (A.11).

Assuming the weights w_{ij}^{mn} are fixed, we solve (A.11) with the Gauss-Newton method. The first frame's pose is held fixed. The cost related to each $\mathbf{T}_m$ is

$$F(\mathbf{T}_m) = \sum_{n \neq m} \sum_{ij} w_{ij}^{mn} (\mathbf{T}_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n)^2 \tag{A.12}$$

We apply a small perturbation $\boldsymbol{\epsilon}_m \in \mathbb{R}^6$ on the right of $\mathbf{T}_m$:

$\mathbf{T}^\star_m = \mathbf{T}_m \exp(\boldsymbol{\epsilon}_m^\wedge) = \mathbf{T}_m \exp(\begin{bmatrix}\rho_m \\ \phi_m\end{bmatrix}^\wedge)$ and then linearize each $w_{ij}^{mn}(\mathbf{T}^\star_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n)$ as

$$\begin{aligned}
w_{ij}^{mn}(\mathbf{T}^\star_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n) &= w_{ij}^{mn}(\mathbf{T}_m \exp(\boldsymbol{\epsilon}^\wedge)\mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n)^2 \\
&\approx w_{ij}^{mn}(\mathbf{T}_m (1 + \boldsymbol{\epsilon}^\wedge)\mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n)^2 \\
&= w_{ij}^{mn}(\mathbf{T}_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n + \mathbf{T}_m \boldsymbol{\epsilon}^\wedge \mathbf{x}_i^m)^2 \\
&= w_{ij}^{mn}(\mathbf{T}_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n + \mathbf{T}_m (\mathbf{x}_i^m)^\odot \boldsymbol{\epsilon})^2 \qquad \text{(A.13)}
\end{aligned}$$

where the $\odot : \mathbb{R}^4 \to \mathbb{R}^{4\times 6}$ operator is defined as

$$\mathbf{x}^\odot = \begin{bmatrix} \mathbf{I} & -\mathbf{x}^\wedge \\ \mathbf{0}^T & \mathbf{0}^T \end{bmatrix} \tag{A.14}$$

The gradient is

$$\begin{aligned}
\frac{\partial}{\partial \boldsymbol{\epsilon}_m} w_{ij}^{mn}(\mathbf{T}^\star_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n) &= 2 w_{ij}^{mn}(\mathbf{T}_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n)^T \mathbf{T}_m (\mathbf{x}_i^m)^\odot \\
&\quad + 2\boldsymbol{\epsilon}^T (\mathbf{T}_m (\mathbf{x}_i^m)^\odot)^T \mathbf{T}_m (\mathbf{x}_i^m)^\odot \\
&:= 2((\mathbf{b}_{ij}^{mn})^T \mathbf{A}_{ij}^{mn} \qquad \text{(A.15)} \\
&\quad + \boldsymbol{\epsilon}^T (\mathbf{A}_{ij}^{mn})^T \mathbf{A}_{ij}^{mn}) \qquad \text{(A.16)}
\end{aligned}$$

where $\mathbf{A}_{ij} = \mathbf{T}_m (\mathbf{x}_i^m)^\odot \in \mathbb{R}^{4\times 6}$ and $\mathbf{b}_{ij}^{mn} = w_{ij}^{mn}(\mathbf{T}_m \mathbf{x}_i^m - \mathbf{T}_n \mathbf{z}_j^n) \in \mathbb{R}^4$. For frame m, we now stack up all the residuals between pairs of points from all frames connected to frame m, and then set it to zero:

$$\mathbf{A}_m = \begin{bmatrix} \mathbf{A}_{11}^{m1} \\ \mathbf{A}_{12}^{m1} \\ \cdots \\ \mathbf{A}_{11}^{m2} \\ \cdots \\ \mathbf{A}_{ij}^{mN} \end{bmatrix}, \quad \mathbf{b}_m = \begin{bmatrix} \mathbf{b}_{11}^{m1} \\ \mathbf{b}_{12}^{m1} \\ \cdots \\ \mathbf{b}_{11}^{m2} \\ \cdots \\ \mathbf{b}_{ij}^{mN} \end{bmatrix} \tag{A.17}$$

$$\Delta \mathbf{T}_m = -(\mathbf{A}_m^T \mathbf{A}_m)^{-1} \mathbf{A}_m^T \mathbf{b}_m \tag{A.18}$$

The corresponding step update for frame m's pose $\mathbf{T}_m$ at the current iteration is

$$\mathbf{T}^\star_m = \mathbf{T}_m \exp(\Delta \mathbf{T}_m) \tag{A.19}$$

BIBLIOGRAPHY

[1] Evan Ackerman. Agility robotics introduces cassie, a dynamic and talented robot delivery ostrich. https://spectrum.ieee.org/agility-robotics-introduces-cassie-a-dynamic-and-talented-robot-delivery-ostrich, 2017.

[2] Pratik Agarwal, Gian Diego Tipaldi, Luciano Spinello, Cyrill Stachniss, and Wolfram Burgard. Robust map optimization using dynamic covariance scaling. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 62–69. Ieee, 2013.

[3] Sameer Agarwal and Keir Mierle. Ceres solver: Tutorial & reference. *Google Inc*, 2(72):8, 2012.

[4] Algobotics. Lane detection demo. https://www.youtube.com/watch?v=KzRkS-8oNtc. Accessed: 2023-01-15.

[5] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. PointNetLK: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 2019.

[6] Xuyang Bai, Zixin Luo, Lei Zhou, Hongbo Fu, Long Quan, and Chiew-Lan Tai. D3feat: Joint learning of dense detection and description of 3D local features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6359–6367, 2020.

[7] Timothy D Barfoot. *State estimation for robotics*. Cambridge University Press, 2024.

[8] Alain Berlinet and Christine Thomas-Agnan. *Reproducing Kernel Hilbert Space in Probability and Statistics*. Springer Science & Business Media, Jan. 2004.

[9] Paul J Besl and Neil D McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, Feb. 1992.

[10] Peter Biber, Sven Fleck, and Wolfgang Straßer. A probabilistic framework for robust and accurate matching of point clouds. In *Joint Pattern Recognition Symposium*, pages 480–487. Springer, 2004.

[11] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003.

[12] 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*, pages 1246–1251, 2005.

[13] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, Berlin, Heidelberg, 2006.

[14] Michael Bloesch, Jan Czarnowski, Ronald Clark, Stefan Leutenegger, and Andrew J Davison. Codeslam—learning a compact, optimisable representation for dense visual slam. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2560–2568, 2018.

[15] Alexandre Boulch, Joris Guerry, Bertrand Le Saux, and Nicolas Audebert. Snapnet: 3d point cloud semantic labeling with 2d deep segmentation networks. *Computers & Graphics*, 71:189 – 198, 2018.

[16] Sean L Bowman, Nikolay Atanasov, Kostas Daniilidis, and George J Pappas. Probabilistic data association for semantic SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1722–1729, 2017.

[17] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[18] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J. Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[19] Dylan Campbell and Lars Petersson. An adaptive data representation for robust point-set registration and merging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4292–4300, 2015.

[20] Carlos Campos, Richard Elvira, Juan J. Gómez Rodríguez, José M. M. Montiel, and Juan D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37:1874–1890, 2021.

[21] J Quinonero Candela. *Learning with uncertainty-Gaussian processes and relevance vector machines*. PhD thesis, Technical University of Denmark, 2004.

[22] Nicholas Carlevaris-Bianco, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *International Journal of Robotics Research*, 35(9):1023–1035, 2016.

[23] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3D point cloud analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 14514–14523, 2021.

[24] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

[25] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European Conference on Computer Vision*, 2018.

[26] Xieyuanli Chen, Thomas Läbe, Andres Milioto, Timo Röhling, Olga Vysotska, Alexandre Haag, Jens Behley, Cyrill Stachniss, and FKIE Fraunhofer. OverlapNet: Loop closing for LiDAR-based SLAM. In *Proceedings of the Robotics: Science and Systems Conference*, 2020.

[27] Xieyuanli Chen, Andres Milioto, Emanuele Palazzolo, Philippe Giguere, Jens Behley, and Cyrill Stachniss. Suma++: Efficient lidar-based semantic SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4530–4537. IEEE, 2019.

[28] Yang Chen and Gérard G Medioni. Object modeling by registration of multiple range images. *Image Vision Comput.*, 10(3):145–155, 1992.

[29] Liang Cheng, Song Chen, Xiaoqiang Liu, Hao Xu, Yang Wu, Manchun Li, and Yanming Chen. Registration of laser scanning point clouds: A review. *Sensors*, 18(5):1641, 2018.

[30] Dmitry Chetverikov, Dmitry Stepanov, and Pavel Krsek. Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm. *Image and Vision Computing*, 23(3):299 – 309, 2005.

[31] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[32] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 8958–8966, 2019.

[33] Haili Chui and Anand Rangarajan. A feature registration framework using mixture models. In *Proceedings IEEE Workshop on Mathematical Methods in Biomedical Image Analysis. MMBIA-2000 (Cat. No. PR00737)*, pages 190–197. IEEE, 2000.

[34] William Clark, Maani Ghaffari, and Anthony Bloch. Nonparametric continuous sensor registration. *Journal of Machine Learning Research*, 22(271):1–50, 2021.

[35] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral CNN. In *Proceedings of the International Conference on Machine Learning*, pages 1321–1330. PMLR, 2019.

[36] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the International Conference on Machine Learning*, pages 2990–2999. PMLR, 2016.

[37] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *Proceedings of the International Conference on Learning Representations*, 2018.

[38] Taco S Cohen and Max Welling. Steerable cnns. *Proceedings of the International Conference on Learning Representations*, 2017.

[39] Open X-Embodiment Collaboration, Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, Antonin Raffin, Ayzaan Wahid, Ben Burgess-Limerick, Beomjoon Kim, Bernhard Schölkopf, Brian Ichter, Cewu Lu, Charles Xu, Chelsea Finn, Chenfeng Xu, Cheng Chi, Chenguang Huang, Christine Chan, Chuer Pan, Chuyuan Fu, Coline Devin, Danny Driess, Deepak Pathak, Dhruv Shah, Dieter Büchler, Dmitry Kalashnikov, Dorsa Sadigh, Edward Johns, Federico Ceola, Fei Xia, Freek Stulp, Gaoyue Zhou, Gaurav S. Sukhatme, Gautam Salhotra, Ge Yan, Giulio Schiavi, Hao Su, Hao-Shu Fang, Haochen Shi, Heni Ben Amor, Henrik I Christensen, Hiroki Furuta, Homer Walke, Hongjie Fang, Igor Mordatch, Ilija Radosavovic, Isabel Leal, Jacky Liang, Jaehyung Kim, Jan Schneider, Jasmine Hsu, Jeannette Bohg, Jeffrey Bingham, Jiajun Wu, Jialin Wu, Jianlan Luo, Jiayuan Gu, Jie Tan, Jihoon Oh, Jitendra Malik, Jonathan Tompson, Jonathan Yang, Joseph J. Lim, João Silvério, Junhyek Han, Kanishka Rao, Karl Pertsch, Karol Hausman, Keegan Go, Keerthana Gopalakrishnan, Ken Goldberg, Kendra Byrne, Kenneth Oslund, Kento Kawaharazuka, Kevin Zhang, Keyvan Majd, Krishan Rana, Krishnan Srinivasan, Lawrence Yunliang Chen, Lerrel Pinto, Liam Tan, Lionel Ott, Lisa Lee, Masayoshi Tomizuka, Maximilian Du, Michael Ahn, Mingtong Zhang, Mingyu Ding, Mohan Kumar Srirama, Mohit Sharma, Moo Jin Kim, Naoaki Kanazawa, Nicklas Hansen, Nicolas Heess, Nikhil J Joshi, Niko Suenderhauf, Norman Di Palo, Nur Muhammad Mahi Shafiullah, Oier Mees, Oliver Kroemer, Pannag R Sanketi, Paul Wohlhart, Peng Xu, Pierre Sermanet, Priya Sundaresan, Quan Vuong, Rafael Rafailov, Ran Tian, Ria Doshi, Roberto Martín-Martín, Russell Mendonca, Rutav Shah, Ryan Hoque, Ryan Julian, Samuel Bustamante, Sean Kirmani, Sergey Levine, Sherry Moore, Shikhar Bahl, Shivin Dass, Shuran Song, Sichun Xu, Siddhant Haldar, Simeon Adebola, Simon Guist, Soroush Nasiriany, Stefan Schaal, Stefan Welker, Stephen Tian, Sudeep Dasari, Suneel Belkhale, Takayuki Osa, Tatsuya Harada, Tatsuya Matsushima, Ted Xiao, Tianhe Yu, Tianli Ding, Todor Davchev, Tony Z. Zhao, Travis Armstrong, Trevor Darrell, Vidhi Jain, Vincent Vanhoucke, Wei Zhan, Wenxuan Zhou, Wolfram Burgard, Xi Chen, Xiaolong Wang, Xinghao Zhu, Xuanlin Li, Yao Lu, Yevgen Chebotar, Yifan Zhou, Yifeng Zhu, Ying Xu, Yixuan Wang, Yonatan Bisk, Yoonyoung Cho, Youngwoon Lee, Yuchen Cui, Yueh hua Wu, Yujin Tang, Yuke Zhu, Yunzhu Li, Yusuke Iwasawa, Yutaka Matsuo, Zhuo Xu, and Zichen Jeff Cui. Open X-Embodiment: Robotic learning datasets and RT-X models. https://arxiv.org/abs/2310.08864, 2023.

[40] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

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

[42] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt.

Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4), 2017.

[43] Martin Danelljan, Giulia Meneghetti, Fahad Shahbaz Khan, and Michael Felsberg. A probabilistic framework for color-based point set registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1818–1826, 2016.

[44] Arun Das and Steven L Waslander. Scan registration using segmented region growing NDT. *International Journal of Robotics Research*, 33(13):1645–1663, 2014.

[45] Andrew J Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 3, pages 1403–1403. IEEE Computer Society, 2003.

[46] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

[47] Pierre Dellenbach, Jean-Emmanuel Deschaud, Bastien Jacquet, and François Goulette. Ct-icp: Real-time elastic lidar odometry with loop closure. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5580–5586. IEEE, 2022.

[48] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for SO(3)-equivariant networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 12200–12209, 2021.

[49] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.

[50] Kevin Doherty, Dehann Fourie, and John Leonard. Multimodal semantic SLAM with probabilistic data association. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2419–2425. IEEE, 2019.

[51] Kevin J Doherty, David P Baxter, Edward Schneeweiss, and John J Leonard. Probabilistic data association via mixture models for robust semantic SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1098–1104. IEEE, 2020.

[52] Kevin J. Doherty, Ziqi Lu, Kurran Singh, and John J. Leonard. Discrete-Continuous Smoothing and Mapping. *IEEE Robotics and Automation Letters*, 7(4):12395–12402, 2022.

[53] Boston Dynamics. Boston dynamics spot. https://bostondynamics.com/products/spot/, 2023.

[54] Ben Eckart, Kihwan Kim, Alejandro Troccoli, Alonzo Kelly, and Jan Kautz. Mlmd: Maximum likelihood mixture decoupling for fast and accurate point cloud registration. In *International Conference on 3D Vision*, pages 241–249. IEEE, 2015.

[55] Benjamin Eckart and Alonzo Kelly. Rem-seg: A robust em algorithm for parallel segmentation and registration of point clouds. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4355–4362. IEEE, 2013.

[56] Benjamin Eckart, Kihwan Kim, and Jan Kautz. Hgmr: Hierarchical Gaussian mixtures for adaptive 3D registration. In *Proceedings of the European Conference on Computer Vision*, pages 705–721, 2018.

[57] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015.

[58] Gil Elbaz, Tamar Avraham, and Anath Fischer. 3d point cloud registration for localization using a deep neural network auto-encoder. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4631–4640, 2017.

[59] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017.

[60] Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *Proceedings of the European Conference on Computer Vision*, pages 834–849. Springer, 2014.

[61] Georgios Dimitrios Evangelidis and Radu Horaud. Joint alignment of multiple point sets with batch and incremental expectation-maximization. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1397–1410, 2017.

[62] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[63] C. Forster, M. Pizzoli, and D. Scaramuzza. Svo: Fast semi-direct monocular visual odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 15–22, 2014.

[64] Dehann Fourie, John Leonard, and Michael Kaess. A nonparametric belief solution to the bayes tree. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2189–2196, 2016.

[65] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. SE(3)-transformers: 3D roto-translation equivariant attention networks. *Proceedings of the Advances in Neural Information Processing Systems Conference*, 33:1970–1981, 2020.

[66] Lu Gan, Ray Zhang, Jessy W. Grizzle, Ryan M. Eustice, and Maani Ghaffari. Bayesian spatial kernel smoothing for scalable dense semantic mapping. *IEEE Robotics and Automation Letters*, 5(2):790–797, 2020.

[67] James Garforth and Barbara Webb. Visual appearance analysis of forest scenes for monocular SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1794–1800. IEEE, 2019.

[68] Daniel Gehrig, Henri Rebecq, Guillermo Gallego, and Davide Scaramuzza. Eklt: Asynchronous photometric feature tracking using events and frames. *International Journal of Computer Vision*, 128(3):601–618, 2020.

[69] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[70] Andreas Geiger, Martin Roser, and Raquel Urtasun. Efficient large-scale stereo matching. In *Proceedings of the Asian Conference on Computer Visio*, 2010.

[71] Maani Ghaffari, William Clark, Anthony Bloch, Ryan M. Eustice, and Jessy W. Grizzle. Continuous direct sparse visual odometry from RGB-D images. In *Proceedings of the Robotics: Science and Systems Conference*, Freiburg, Germany, June 2019.

[72] Ross Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[73] Zan Gojcic, Caifa Zhou, Jan D Wegner, and Andreas Wieser. The perfect match: 3d point cloud matching with smoothed densities. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5545–5554, 2019.

[74] Steven Gold, Anand Rangarajan, Chien-Ping Lu, Suguna Pappu, and Eric Mjolsness. New algorithms for 2d and 3d point matching: Pose estimation and correspondence. *Pattern recognition*, 31(8):1019–1031, 1998.

[75] Jacob Goldberger. Registration of multiple point sets using the em algorithm. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2, pages 730–736. IEEE, 1999.

[76] Sébastien Granger and Xavier Pennec. Multi-scale EM-ICP: A fast and robust approach for surface registration. In *Proceedings of the European Conference on Computer Vision*, pages 418–432, 2002.

[77] Giorgio Grisetti, Rainer Kümmerle, Hauke Strasdat, and Kurt Konolige. g2o: A general framework for (hyper) graph optimization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 9–13, 2011.

[78] Michael Grupp. evo: Python package for the evaluation of odometry and slam. https://github.com/MichaelGrupp/evo, 2017.

[79] Timo Hackel, N. Savinov, L. Ladicky, Jan D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pages 91–98, 2017.

[80] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[81] Hu He and Ben Upcroft. Nonparametric semantic segmentation for 3D street scenes. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3697–3703, 2013.

[82] Alexander Hermans, Georgios Floros, and Bastian Leibe. Dense 3D semantic mapping of indoor scenes from RGB-D images. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2631–2638, 2014.

[83] Radu Horaud, Florence Forbes, Manuel Yguel, Guillaume Dewaele, and Jian Zhang. Rigid and articulated point registration with expectation conditional maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):587–602, 2010.

[84] Armin Hornung, Kai M Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 34(3):189–206, 2013.

[85] Gibson Hu, Kasra Khosoussi, and Shoudong Huang. Towards a reliable SLAM back-end. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 37–43. IEEE, 2013.

[86] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, Andreas Wieser, and Konrad Schindler. Predator: Registration of 3D point clouds with low overlap. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4267–4276, 2021.

[87] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. ACM sym. User interface software and tech.*, pages 559–568. ACM, 2011.

[88] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, and Andrew 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. ACM, 2011.

[89] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[90] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3304–3311. IEEE, 2010.

[91] B. Jian and B. C. Vemuri. Robust point set registration using Gaussian mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, Aug 2011.

[92] Michael Kaess, Hordur Johannsson, Richard Roberts, Viorela Ila, John J Leonard, and Frank Dellaert. isam2: Incremental smoothing and mapping using the bayes tree. *International Journal of Robotics Research*, 31(2):216–235, 2012.

[93] Benjamin Katz, Jared Di Carlo, and Sangbae Kim. Mini cheetah: A platform for pushing the limits of dynamic quadruped control. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 6295–6301. IEEE, 2019.

[94] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4), 2023.

[95] Christian Kerl. Dense Visual Odometry (DVO) Code. https://github.com/tum-vision/dvo, 2013.

[96] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Dense visual slam for RGB-D cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2100–2106. IEEE, 2013.

[97] Christian Kerl, Jürgen Sturm, and Daniel Cremers. Robust odometry estimation for RGB-D cameras. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3748–3754. IEEE, 2013.

[98] Byung-soo Kim, Pushmeet Kohli, and Silvio Savarese. 3D scene understanding by voxel-CRF. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1425–1432, 2013.

[99] David M Knigge, David W Romero, and Erik J Bekkers. Exploiting redundancy: Separable group convolutional networks on lie groups. In *Proceedings of the International Conference on Machine Learning*, pages 11359–11386. PMLR, 2022.

[100] Lukas Koestler, Nan Yang, Niclas Zeller, and Daniel Cremers. Tandem: Tracking and dense mapping in real-time using deep multi-view stereo. In *Conference on Robot Learning*, pages 34–45. PMLR, 2022.

[101] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Globally consistent 3d lidar mapping with gpu-accelerated gicp matching cost factors. *IEEE Robotics and Automation Letters*, 6(4):8591–8598, 2021.

[102] Loïc Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4558–4567, 2018.

[103] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Proceedings of the Advances in Neural Information Processing Systems Conference*, pages 820–830, 2018.

[104] Tzu-Yuan Lin, William Clark, Ryan M Eustice, Jessy W Grizzle, Anthony Bloch, and Maani Ghaffari. Adaptive continuous visual odometry from RGB-D images. *arXiv preprint arXiv:1910.00713*, 2019.

[105] Xiyuan Liu, Zheng Liu, Fanze Kong, and Fu Zhang. Large-scale lidar consistent mapping using hierarchical lidar bundle adjustment. *IEEE Robotics and Automation Letters*, 8(3):1523–1530, 2023.

[106] Zheng Liu and Fu Zhang. Balm: Bundle adjustment for lidar mapping. *IEEE Robotics and Automation Letters*, 6(2):3184–3191, 2021.

[107] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[108] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004.

[109] W. Lu, Guowei Wan, Y. Zhou, Xiangyu Fu, P. Yuan, and Shiyu Song. Deepvcp: An end-to-end deep neural network for point cloud registration. *Proceedings of the IEEE International Conference on Computer Vision*, pages 12–21, 2019.

[110] Lachlan E MacDonald, Sameera Ramasinghe, and Simon Lucey. Enabling equivariance for arbitrary lie groups. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8183–8192, 2022.

[111] Martin Magnusson, Achim Lilienthal, and Tom Duckett. Scan registration for autonomous mining vehicles using 3D-NDT. *Journal of Field Robotics*, 24(10):803–827, 2007.

[112] Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. Point registration via efficient convex relaxation. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016.

[113] Daniel Maturana. scrollgrid. `https://github.com/dimatura/scrollgrid`, 2016.

[114] Yanzi Miao, Yang Liu, Hongbin Ma, and Huijie Jin. The pose estimation of mobile robot based on improved point cloud registration. *International Journal of Advanced Robotic Systems*, 13(2):52, 2016.

[115] Zhe Min, Jiaole Wang, and Max Q-H Meng. Joint rigid registration of multiple generalized point sets with hybrid mixture models. *IEEE Transactions on Automation Science and Engineering*, 17(1):334–347, 2019.

[116] Zhixiang Min, Yiding Yang, and Enrique Dunn. Voldor: Visual odometry from log-logistic dense optical flow residuals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4898–4909, 2020.

[117] Niloy J. Mitra, Natasha Gelfand, Helmut Pottmann, and Leonidas Guibas. Registration of point cloud data from a geometric optimization perspective. In *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, SGP '04, page 22–31, New York, NY, USA, 2004. Association for Computing Machinery.

[118] Beipeng Mu, Shih-Yuan Liu, Liam Paull, John Leonard, and Jonathan P How. SLAM with objects using a nonparametric pose graph. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4602–4609. IEEE, 2016.

[119] Marius Muja and David Lowe. Flann-fast library for approximate nearest neighbors user manual. *Computer Science Department, University of British Columbia, Vancouver, BC, Canada*, 5, 2009.

[120] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.

[121] Raúl Mur-Artal and Juan D. Tardós. ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras. *IEEE Transactions on Robotics*, 33(5):1255–1262, 2017.

[122] Kevin P Murphy. *Machine learning: a probabilistic perspective*. The MIT Press, 2012.

[123] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from RGBD images. In *Proceedings of the European Conference on Computer Vision*, 2012.

[124] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer New York, 1996.

[125] Richard A Newcombe, Steven J Lovegrove, and Andrew J Davison. DTAM: Dense tracking and mapping in real-time. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2320–2327. IEEE, 2011.

[126] University of Michigan Campus Information. The wave field on north. `https://campusinfo.umich.edu/article/wave-field-north`, 2023.

[127] Edwin Olson and Pratik Agarwal. Inference on networks of mixtures for robust robot mapping. *International Journal of Robotics Research*, 32(7):826–840, 2013.

[128] Joseph Ortiz, Talfan Evans, Edgar Sucar, and Andrew J. Davison. Incremental abstraction in distributed probabilistic SLAM graphs. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2022.

[129] G. D. Pais, S. Ramalingam, V. M. Govindu, J. C. Nascimento, R. Chellappa, and P. Miraldo. 3DRegNet: A deep neural network for 3d point registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7191–7201, 2020.

[130] Yue Pan, Pengchuan Xiao, Yujie He, Zhenlei Shao, and Zesong Li. Mulls: Versatile lidar slam via multi-metric linear least square. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 11633–11640. IEEE, 2021.

[131] Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Colored point cloud registration revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 143–152, 2017.

[132] Steven A Parkison, Lu Gan, Maani Ghaffari Jadidi, and Ryan M Eustice. Semantic iterative closest point through expectation-maximization. In *Proceedings of the British Machine Vision Conference*, page 280, 2018.

[133] Steven A Parkison, Maani Ghaffari, Lu Gan, Ray Zhang, Arash K Ushani, and Ryan M Eustice. Boosting shape registration algorithms via reproducing kernel Hilbert space regularizers. *IEEE Robotics and Automation Letters*, 4(4):4563–4570, 2019.

[134] Tyson Govan Phillips, Nicky Guenther, and Peter Ross McAree. When the dust settles: The four behaviors of lidar in the presence of fine airborne particulates. *Journal of Field Robotics*, 34(5):985–1009, 2017.

[135] Luis Pineda, Taosha Fan, Maurizio Monge, Shobha Venkataraman, Paloma Sodhi, Ricky TQ Chen, Joseph Ortiz, Daniel DeTone, Austin Wang, Stuart Anderson, et al. Theseus: A library for differentiable nonlinear optimization. *Proceedings of the Advances in Neural Information Processing Systems Conference*, 35:3801–3818, 2022.

[136] Matthieu Pizenberg. DVO (without ROS dependency) code. https://github.com/mpizenberg/dvo/tree/76f65f0c9b438675997f595471d39863901556a9, 2019.

[137] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015.

[138] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from RGB-D data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[139] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[140] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Proceedings of the Advances in Neural Information Processing Systems Conference*, pages 5099–5108, 2017.

[141] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, Slobodan Ilic, Dewen Hu, and Kai Xu. Geotransformer: Fast and robust point cloud registration with geometric transformer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[142] Anand Rangarajan, Haili Chui, and James S Duncan. Rigid point feature registration using mutual information. *Medical Image Analysis*, 3(4):425–440, 1999.

[143] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*, volume 1. MIT press, 2006.

[144] Andrea Romanoni and Matteo Matteucci. Tapa-mvs: Textureless-aware patchmatch multi-view stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10413–10422, 2019.

[145] David M Rosen, Kevin J Doherty, Antonio Terán Espinoza, and John J Leonard. Advances in inference and representation for simultaneous localization and mapping. *Annual Review of Control, Robotics, and Autonomous Systems*, 4:215–242, 2021.

[146] Antoni Rosinol, John J Leonard, and Luca Carlone. Nerf-slam: Real-time dense monocular slam with neural radiance fields. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3437–3444. IEEE, 2023.

[147] Edward Rosten and Tom Drummond. Machine learning for high-speed corner detection. In *Proceedings of the European Conference on Computer Vision*, pages 430–443. Springer, 2006.

[148] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2564–2571. Ieee, 2011.

[149] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3):157–173, 2008.

[150] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3212–3217. IEEE, 2009.

[151] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *Proceedings of the IEEE International Conference on Robotics and Automation*, Shanghai, China, May 9-13 2011.

[152] Renato F. Salas-Moreno, Richard A. Newcombe, Hauke Strasdat, Paul H.J. Kelly, and Andrew J. Davison. SLAM++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1352–1359, 2013.

[153] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[154] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4938–4947, 2020.

[155] Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E(n) equivariant graph neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 9323–9332. PMLR, 2021.

[156] Davide Scaramuzza and Friedrich Fraundorfer. Visual odometry [tutorial]. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011.

[157] Bernhard Schölkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[158] Thomas Schops, Torsten Sattler, and Marc Pollefeys. BAD SLAM: Bundle adjusted direct RGB-D SLAM. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 134–144, 2019.

[159] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-ICP. In *Proceedings of the Robotics: Science and Systems Conference*, volume 2 Issue 4, page 435. Seattle, WA, 2009.

[160] Anuj Sehgal, Daniel Cernea, and Milena Makaveeva. Real-time scale invariant 3d range point cloud registration. In *Proceedings of the International Conference on Image Analysis and Recognition*, pages 220–229, 06 2010.

[161] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip HS Torr. Urban 3D semantic modelling using stereo vision. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 580–585, 2013.

[162] James Servos and Steven L Waslander. Multi channel generalized-ICP. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3644–3649. IEEE, 2014.

[163] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4758–4765. IEEE, 2018.

[164] Gregory C Sharp, Sang W Lee, and David K Wehe. ICP registration using invariant features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1):90–102, 2002.

[165] Hauke Strasdat, J Montiel, and Andrew J Davison. Scale drift-aware large scale monocular slam. *Robotics: Science and Systems VI*, 2(3):7, 2010.

[166] Jörg Stückler, Nenad Biresev, and Sven Behnke. Semantic mapping using object-class segmentation of RGB-D images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3005–3010, 2012.

[167] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[168] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018.

[169] Edgar Sucar, Shikun Liu, Joseph Ortiz, and Andrew J Davison. imap: Implicit mapping and positioning in real-time. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6229–6238, 2021.

[170] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett. Recurrent-octomap: Learning state-based map refinement for long-term semantic mapping with 3-d-lidar data. *IEEE Robotics and Automation Letters*, 3(4):3749–3756, Oct 2018.

[171] Niko Sünderhauf and Peter Protzel. Switchable constraints for robust pose graph SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1879–1884. IEEE, 2012.

[172] Chengzhou Tang and Ping Tan. Ba-net: Dense bundle adjustment network. *Proceedings of the International Conference on Learning Representations*, 2019.

[173] Jigang Tang, Songbin Li, and Peng Liu. A review of lane detection methods based on deep learning. *Pattern Recognition*, 111:107623, 2021.

[174] Keisuke Tateno, Federico Tombari, Iro Laina, and Nassir Navab. Cnn-slam: Real-time dense monocular slam with learned depth prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6243–6252, 2017.

[175] NASA Mars Communications Team. Moving around mars. https://mars.nasa.gov/mer/mission/timeline/surfaceops/navigation/. Accessed: 2023-01-15.

[176] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Proceedings of the European Conference on Computer Vision*, pages 402–419. Springer, 2020.

[177] Zachary Teed and Jia Deng. Droid-slam: Deep visual slam for monocular, stereo, and RGB-D cameras. *Advances in neural information processing systems*, 34:16558–16569, 2021.

[178] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6411–6420, 2019.

[179] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3D point clouds. *arXiv preprint arXiv:1802.08219*, 2018.

[180] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9627–9636, 2019.

[181] Michael E Tipping. Sparse Bayesian learning and the relevance vector machine. *J. machine learning res.*, 1(Jun):211–244, 2001.

[182] Michael E Tipping, Anita C Faul, et al. Fast marginal likelihood maximisation for sparse bayesian models. In *AISTATS*, 2003.

[183] Shaozu Cao Tong Qin. A-loam code. `https://github.com/HKUST-Aerial-Robotics/A-LOAM`, 2019.

[184] Bill Triggs, Philip F McLauchlan, Richard I Hartley, and Andrew W Fitzgibbon. Bundle adjustment—a modern synthesis. In *Vision Algorithms: Theory and Practice: International Workshop on Vision Algorithms*, pages 298–372. Springer, 2000.

[185] Yanghai Tsin and Takeo Kanade. A correlation-based approach to robust point set registration. In *Proceedings of the European Conference on Computer Vision*, pages 558–569. Springer, 2004.

[186] Greg Turk. The stanford 3d scanning repository. `https://graphics.stanford.edu/data/3Dscanrep/`, 2000.

[187] Tommi Tykkälä and Andrew I Comport. A dense structure model for image based stereo SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1758–1763. IEEE, 2011.

[188] Cihan Ulaş and Hakan Temeltaş. 3D multi-layered normal distribution transform for fast and long range scan matching. *Journal of Intelligent & Robotic Systems*, 71(1):85–108, 2013.

[189] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[190] George Vogiatzis and Carlos Hernández. Video-based, real-time multi-view stereo. *Image and Vision Computing*, 29(7):434–441, 2011.

[191] Fei Wang, Baba C Vemuri, and Anand Rangarajan. Groupwise point pattern registration using a novel cdf-based jensen-shannon divergence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1283–1288. IEEE, 2006.

[192] Fei Wang, Baba C Vemuri, Anand Rangarajan, and Stephan J Eisenschenk. Simultaneous nonrigid registration of multiple point sets and atlas construction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(11):2011–2022, 2008.

[193] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3903–3911, 2017.

[194] Wenshan Wang, Delong Zhu, Xiangwei Wang, Yaoyu Hu, Yuheng Qiu, Chen Wang, Yafei Hu, Ashish Kapoor, and Sebastian Scherer. Tartanair: A dataset to push the limits of visual slam. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4909–4916. IEEE, 2020.

[195] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3523–3532, 2019.

[196] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (TOG)*, 2019.

[197] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. Nerf–: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[198] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. *Proceedings of the Advances in Neural Information Processing Systems Conference*, 31, 2018.

[199] Lorenz Wellhausen, René Ranftl, and Marco Hutter. Safe robot navigation via multi-modal anomaly detection. *IEEE Robotics and Automation Letters*, 5(2):1326–1333, 2020.

[200] P. Wenzel, R. Wang, N. Yang, Q. Cheng, Q. Khan, L. von Stumberg, N. Zeller, and D. Cremers. 4Seasons: A cross-season dataset for multi-weather SLAM in autonomous driving. In *Proceedings of the German Conference on Pattern Recognition (GCPR)*, 2020.

[201] Thomas Whelan, Hordur Johannsson, Michael Kaess, John J Leonard, and John McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 5724–5731. IEEE, 2013.

[202] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[203] Wooptix. Illustration of stereo matching. `https://wooptix.com/drawbacks-of-widespread-stereo-matching-techniques/`, 2023.

[204] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. SqueezeSeg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3D LiDAR point cloud. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1887–1893. IEEE, 2018.

[205] Fang Wu and Giovanni Beltrame. Direct sparse odometry with planes. *IEEE Robotics and Automation Letters*, pages 1–1, 2021.

[206] Jiatian Wu. Direct Sparse Odometry with Stereo Cameras code. `https://github.com/JiatianWu/stereo-dso`, 2023.

[207] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015.

[208] Yu Xiang and Dieter Fox. DA-RNN: Semantic mapping with data associated recurrent neural networks. In *Proceedings of the Robotics: Science and Systems Conference*, 2017.

[209] Haofei Xu, Jing Zhang, Jianfei Cai, Hamid Rezatofighi, Fisher Yu, Dacheng Tao, and Andreas Geiger. Unifying flow, stereo and depth estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[210] Heng Yang, Jingnan Shi, and Luca Carlone. Teaser: Fast and certifiable point cloud registration. *IEEE Transactions on Robotics*, 37(2):314–333, 2020.

[211] Nan Yang, Lukas von Stumberg, Rui Wang, and Daniel Cremers. D3vo: Deep depth, deep pose and deep uncertainty for monocular visual odometry. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1281–1292, 2020.

[212] Nan Yang, Rui Wang, Jorg Stuckler, and Daniel Cremers. Deep virtual stereo odometry: Leveraging deep depth prediction for monocular direct sparse odometry. In *Proceedings of the European Conference on Computer Vision*, pages 817–833, 2018.

[213] Shichao Yang, Yulan Huang, and Sebastian Scherer. Semantic 3D occupancy mapping through efficient high order CRFs. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 590–597, 2017.

[214] Shichao Yang and Sebastian Scherer. Cubeslam: Monocular 3-D object SLAM. *IEEE Transactions on Robotics*, 35(4):925–938, 2019.

[215] Zi Jian Yew and Gim Hee Lee. 3dfeat-net: Weakly supervised local 3d features for point cloud registration. In *Proceedings of the European Conference on Computer Vision*, pages 630–646. Springer, 2018.

[216] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. In *Proceedings of the International Conference on Learning Representations*, pages 1–13, Vancouver, BC, Canada, 2016.

[217] Hao Yu, Fu Li, Mahdi Saleh, Benjamin Busam, and Slobodan Ilic. Cofinet: Reliable coarse-to-fine correspondences for robust pointcloud registration. *Proceedings of the Advances in Neural Information Processing Systems Conference*, 34:23872–23884, 2021.

[218] Christopher Zach. Robust bundle adjustment revisited. In *Proceedings of the European Conference on Computer Vision*, pages 772–787. Springer, 2014.

[219] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from RGB-D reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017.

[220] Ji Zhang and Sanjiv Singh. LOAM: Lidar odometry and mapping in real-time. In *Proceedings of the Robotics: Science and Systems Conference*, volume 2 Issue 9, pages 1–9, 2014.

[221] Jianbo Zhang, Liang Yuan, Teng Ran, Qing Tao, and Li He. Bayesian nonparametric object association for semantic SLAM. *IEEE Robotics and Automation Letters*, 6(3):5493–5500, 2021.

[222] Ray Zhang. Unifiedcvo code. `https://github.com/UMich-CURLY/unified_cvo`, 2020.

[223] Ray Zhang, Tzu-Yuan Lin, Chien-Erh Lin, Steven A. Parkison, William Clark, Jessy W. Grizzle, Ryan M. Eustice, and Maani Ghaffari. A new framework for registration of semantic point clouds from stereo and RGB-D cameras. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 12214–12221, 2020.

[224] Xiaoyu Zhang, Wei Wang, Xianyu Qi, Ziwei Liao, and Ran Wei. Point-plane SLAM using supposed planes for indoor environments. *Sensors*, 19(17), 2019.

[225] Zhe Zhao and Xiaoping Chen. Building 3D semantic maps for mobile robots using RGB-D camera. *Intell. Serv. Robot.*, 9(4):297–309, 2016.

[226] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *Proceedings of the European Conference on Computer Vision*, pages 766–782. Springer, 2016.

[227] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018.

[228] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1851–1858, 2017.

[229] Minghan Zhu, Maani Ghaffari, William A Clark, and Huei Peng. E2PN: Efficient SE(3)-equivariant point network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1223–1232, 2023.

[230] Minghan Zhu, Maani Ghaffari, and Huei Peng. Correspondence-free point cloud registration with SO(3)-equivariant implicit shape representations. In *Conference on Robot Learning*, pages 1412–1422. PMLR, 2022.

[231] Yi Zhu, Karan Sapra, Fitsum A Reda, Kevin J Shih, Shawn Newsam, Andrew Tao, and Bryan Catanzaro. Improving semantic segmentation via video propagation and label relaxation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8856–8865, 2019.

[232] Jon Zubizarreta, Iker Aguinaga, and Jose Maria Martinez Montiel. Direct sparse mapping. *IEEE Transactions on Robotics*, 36(4):1363–1370, 2020.