Boosting Shape Registration Algorithms via Reproducing Kernel Hilbert Space Regularizers

Steven A. Parkison, Maani Ghaffari, Lu Gan, Ray Zhang, Arash K. Ushani, and Ryan M. Eustice

***Abstract*—The essence of most shape registration algorithms is to find correspondences between two point clouds and then to solve for a rigid body transformation that aligns the geometry. The main drawback is that the point clouds are obtained by placing the sensor at different views; consequently, the two matched points most likely do not correspond to the same physical point in the real environment. In other words, the point cloud is a discrete representation of the shape geometry. Alternatively, a point cloud measurement can be seen as samples from geometry, and a function can be learned for a continuous representation using regression techniques such as kernel methods. To boost registration algorithms, this work develops a novel class of regularizers modeled in the Reproducing Kernel Hilbert Space (RKHS) that ensures correspondences are also consistent in an abstract vector space of functions such as intensity surface. Furthermore, the proposed RKHS regularizer is agnostic to the choice of the registration cost function which is desirable. The evaluations on experimental data confirm the effectiveness of the proposed regularizer using RGB-D and LIDAR sensors.**

***Index Terms*—Range Sensing, RGB-D Perception, Localization.**

I. Introduction

THE shape registration problem is formulated as finding a *rigid body transformation* that aligns a set of *source points* to a set of *target points*. Registration algorithms can be divided into coarse alignment methods and fine alignment methods. Coarse alignment methods usually do not assume large overlap nor need an initial transformation, but only achieve a crude registration. In [1], a Fourier-based method is proposed to estimate the rotation of limited overlap point clouds. Recently, a deep neural network is used to encode local 3D geometric structures for coarse registration [2]. This paper focuses on the fine registration problem.

Most modern fine alignment methods are derived from the Iterative Closest Point (ICP) algorithm, developed in [3]. ICP iterates between finding the closest pair of points between the two sets of points, and minimizing the sum of geometric residuals between them. The ICP algorithm is extended to minimize point to line [4], and point to plane residuals [5].

These geometric interpretations of the registration problem have been extended to probabilistic frameworks. In the Generalized ICP (GICP) algorithm [6], a Gaussian distribution is fit to the neighboring points of every point in each point cloud. Meanwhile, pairs of points whose residuals go beyond a hard threshold will be discarded. The Normal Distribution Transform (NDT) [7] divides $\mathbb{R}^3$ into voxels and fits a Gaussian distribution to all the points that fall into each voxel. Both algorithms then minimize the (Gaussian) distribution-to-distribution distance between the *target* and the *source* point clouds.

Purely geometric registration methods, such as the ones mentioned previously, ignore additional information obtained by the sensors, such as *RGB* (color information) or *intensity* values. We propose to use these additional channels of information to regularize the inherently geometric registration problem. This idea is not new and others have used RGB

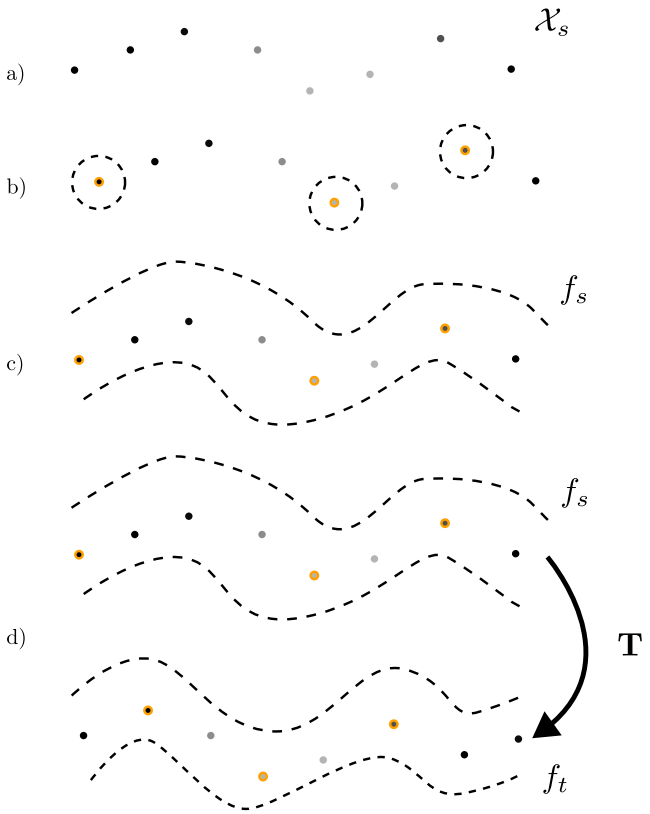

Fig. 1: An illustration of the proposed regularization method. a) Shows the source point cloud $\mathcal{X}_s$ with intensity values. b) illustrates the first stage of our approach in which we train relevance vectors to approximate the intensity function f_s in c). Finally in d) we minimize the regularized cost function to find the optimal transformation $\mathbf{T}$. We argue that the proposed RKHS regularizer is a natural regularizer for the registration problem at hand as it is agnostic to the choice of the registration cost function and is applicable to both LIDAR and RGB-D camera measurements.

Manuscript received: January, 20th, 2019; Revised June, 18th, 2019; Revised April, 19th, 2019; Accepted July, 14th, 2019.

This paper was recommended for publication by Editor C. Stachniss upon evaluation of the Associate Editor and Reviewers' comments.

The authors are with the University of Michigan, Ann Arbor, MI 48109 USA {sparki, maanigj, ganlu, rzh, aushani, eustice}@umich.edu.

Digital Object Identifier (DOI): see top of this page.

and intensity channels in the registration problem. In [8], color is added to the nearest neighbor search to find the point minimizing the 6D distance, and then the 3D geometric distance between those points is minimized. Color Supported GICP [9] also adds color to the nearest neighbor search but defines the distance using the CIELAB color space instead of RGB. In Multi-Chanel GICP [10] a Gaussian distribution is fit over the Euclidean position parameters and the color channels of each point and the same distribution-to-distribution cost function in [6] is minimized. It also uses the extra channels in association by constructing a higher dimensional KD-tree. In Color-NDT [11] a *Gaussian mixture model* is constructed from the color channels of the points that fall into a voxel.

In the application of autonomous vehicles, intensity values from LIDAR have been used for online localization of a platform vehicle through registration [12], [13]. First, an orthographic map of LIDAR intensity is generated as *a priori* using a Simultaneous Localization and Mapping (SLAM) pipeline. Then, during online operation, intensity observations from LIDAR are used to register current observations into the map prior. This idea can be generalized for applications with other sensory modalities, such as cameras [14].

Recent work towards direct visual odometry has led to a different approach to the registration problem using color or intensity based on the photo-consistency assumption. Instead of minimizing geometric residuals, these methods minimize *photometric errors* [15], [16]. This is done by reprojecting the source points into the image frame in which the target points were captured and then minimizing the difference between RGB or intensity values. However, outliers caused by brightness changes do exist across different frames. In [15], a customized sensor model, t-distribution, is introduced to model the error distribution, compensating the frequency of very large or very small photometric residuals.

Optical flow and scene flow approaches use similar methods to estimate relative motion. In optical flow, pixel-wise 2D relative motion between a pair of images can be estimated by leveraging an appearance-based constancy metric, such as brightness constancy, in an energy-minimization framework [17]–[21]. Scene flow tackles a similar problem in 3D with the use of a stereo camera system or active depth sensing [22]–[27].

Several methods have been developed that use the semantic label output of classifiers using intensity and RGBD point clouds as input. In [28] semantic-assisted NDT, which restricts associates to points in the same class, and semantic-assisted GICP, which also restricts associations to the same class and computes the local covariance used in GICP using points of the same class, were introduced. A soft approach based on GICP and using expectation maximization with semantic probability distributions for associations was introduced in [29].

A. Contributions

We develop a novel class of regularizers modeled in the Reproducing Kernel Hilbert Space (RKHS) that ensures correspondences are also consistent in an abstract vector space of functions such as intensity surface, illustrated in Figure 1. The contributions of this work are as follows:

1) assuming the local consistency of point cloud intensity, we develop a class of regularizers to the Generalized-ICP registration algorithm over $\mathrm{SE}(3)$. To account for possible mismatches during data association, instead of using the difference of intensity directly, we learn the point cloud intensity function from noisy intensity measurements;
2) the open source implementation of the developed method including the registration and regression algorithms[1]
3) we evaluate the proposed method using publicly available experimental data and show the performance relative to related baselines.

B. Outline

Section II provides the required preliminaries and notation. The problem formulation is given in Section III. Section IV discusses our main result on a class of regularized shape registration algorithms and an instance of RKHS regularization via sparse Bayesian inference algorithms. Section V presents the empirical results on LIDAR and RGB-D sensor data. Finally, Section VI concludes the paper and discusses future research directions.

II. Mathematical Preliminaries and Notation

Details of the covered topics in this section are available in [30]–[33]. We denote $\|\mathbf{e}\|^2_{\mathbf{\Sigma}} \triangleq \mathbf{e}^\mathsf{T}\mathbf{\Sigma}^{-1}\mathbf{e}$, $\|\cdot\|$ denotes the Euclidean norm, and $\|\mathbf{A}\|^2_\mathrm{F} = \mathrm{Tr}(\mathbf{A}^\mathsf{T}\mathbf{A})$ is the Frobenius norm. The $n \times n$ identity matrix and the n-vector of zeros are denoted $\mathbf{I}_n$ and $\mathbf{0}_n$, respectively. The vector constructed by stacking x_i, $\forall\ i \in \{1, \ldots, n\}$ is denoted $\mathrm{vec}(x_1, \ldots, x_n)$.

A. Matrix Group of Motion in $\mathbb{R}^3$

The general linear group of degree n, denoted $\mathrm{GL}_n(\mathbb{R})$, is the set of all $n \times n$ real invertible matrices, where the group binary operation is the ordinary matrix multiplication. The 3D special orthogonal group, denoted

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

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

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

is the group of rigid transformations on $\mathbb{R}^3$. Let $\hat{\mathbf{T}} \in \mathrm{SE}(3)$ be an estimate of the true transformation $\mathbf{T} \in \mathrm{SE}(3)$. We can compute the rotational (misalignment angle) and translational distances (errors) using $\|\log_\mathrm{m}(\hat{\mathbf{R}}\mathbf{R}^\mathsf{T})\|_\mathrm{F}$ and $\|\hat{\mathbf{p}} - \hat{\mathbf{R}}\mathbf{R}^\mathsf{T}\mathbf{p}\|$, respectively, where $\log_\mathrm{m}(\cdot)$ computes the matrix logarithm. Consistently, the transformation distance that can be directly computed using $\mathrm{d}(\mathbf{T}_1, \mathbf{T}_2) \triangleq \|\log_\mathrm{m}(\hat{\mathbf{T}}\mathbf{T}^{-1})\|_\mathrm{F}$.

B. Representation and Reproducing Kernel Hilbert Space

A Hilbert space is a complete inner product space. Let $(\mathcal{H}, \langle\cdot,\cdot\rangle_\mathcal{H})$ be a real Hilbert space of functions with the inner

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

product between any two square-integrable functions $f, g \in \mathcal{H}$ (or $f, g \in L^2(\mathbb{R}, \mu)$) defined as:

$$\langle f, g \rangle_{\mathcal{H}} \triangleq \int f(\mathbf{x}) g(\mathbf{x}) \mathrm{d}\mu(\mathbf{x}), \tag{1}$$

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

Definition 1 (Reproducing Kernel Hilbert Space [31])**.** *Let $\mathcal{H}$ be a real-valued Hilbert space on a non-empty set $\mathcal{X}$. A function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a reproducing kernel of the Hilbert space $\mathcal{H}$ iff:*

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

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

The second property is called *the reproducing property*; that is using the inner product of f with $k(\cdot, \mathbf{x})$, the value of function f is reproduced at point $\mathbf{x}$. There is a one-to-one relation between a reproducing kernel and its associated RKHS, and such a reproducing kernel is unique [31]. Therefore, our problem reduces to finding an appropriate kernel.

Finally, the nonparametric representer theorem [34] ensures that the solution of minimizing the regularized risk functional admits a representation of the form

$$f(\cdot) = \sum_{i=1}^{m} \alpha_i k(\cdot, \mathbf{x}_i). \tag{2}$$

III. Problem Statement and Formulation

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

Definition 2 (Target point cloud)**.** *The point cloud $\mathcal{X}_t$ which is considered to be in a fixed reference frame is called the target point cloud.*

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

Definition 4 (Target function)**.** *Let $\mathcal{X}_t \subset \mathbb{R}^3$ be a point cloud which is considered in the fixed reference frame. The function $f_t : \mathcal{X}_t \to \mathbb{R}$ is called the target function.*

Definition 5 (Source function)**.** *Let $\mathcal{X}_s \subset \mathbb{R}^3$ be a point cloud which $\mathbf{T} \in \mathrm{SE}(3)$ acts on it. The function $f_s : \mathcal{X}_s \to \mathbb{R}$ is called the source function.*

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

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

The ICP approach follows an iterative two-step procedure for solving the point cloud registration problem: 1) determine the association $\mathcal{I}$ using a Nearest Neighbor (NN) search; 2) minimize the cost defined using the residual, $\mathcal{R}$, over the parameter $\mathbf{T}$. In this work, we use a variant of GICP [6] that we call GICP-SE(3) [29]; the main difference of GICP-SE(3) is solving the optimization on $\mathrm{SE}(3)$ rather than Euler angles parametrization which improves the convergence, and using a Cauchy loss function for robust estimation which removes the need for setting a commonly used distance threshold to accept or reject nearest neighbor data associations.

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

$$\|f_t - f_s\|_{\mathcal{H}_k}^2 = \left(f_t(\mathbf{T} \cdot \mathbf{x}_k^s) - f_s(\mathbf{x}_k^s)\right)^2. \tag{3}$$

Adding this equality constraint to the original problem and using the method of Lagrange multipliers, we arrive at the regularized shape registration problem, as shown in Problem 1. Further, we define the regularizer term as

$$\mathsf{reg}(\mathbf{T}) \triangleq \lambda \sum_{k=1}^{n} \left(f_t(\mathbf{T} \cdot \mathbf{x}_k^s) - f_s(\mathbf{x}_k^s)\right)^2. \tag{4}$$

Problem 1 ($\mathcal{H}_k$-regularized shape registration)**.** *Let $\mathcal{X}_t$ and $\mathcal{X}_s$ be two geometric point clouds and f_t and f_s be target and source functions learned using intensity measurements of their corresponding point clouds, respectively. Given correspondences between target and source point clouds, the optimal transformation that aligns source to target can be computed by solving the following regularized Maximum Likelihood Estimation (MLE) problem:*

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

IV. A Class of $\mathcal{H}_k$-Regularized Shape Registration Algorithms

We model measurements in the target and source clouds as being drawn from Gaussian distributions, i.e.,

$\mathbf{x}_k^t \sim \mathcal{N}(\hat{\mathbf{x}}_k^t, \mathbf{\Sigma}_k^t)$, and $\mathbf{x}_k^s \sim \mathcal{N}(\mathbf{T} \cdot \hat{\mathbf{x}}_k^s, \mathbf{\Sigma}_k^s)$, respectively. Therefore, the residual log-likelihood, excluding the normalization constant, becomes:

$$\mathsf{cost}(\mathbf{T}) = \sum_{k=1}^{n} \|\mathbf{x}_k^t - \mathbf{T} \cdot \mathbf{x}_k^s\|^2_{\mathbf{C}_k}, \tag{5}$$

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

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

The original GICP [6] removes residuals larger than a certain value to ensure that any point in the source cloud which does not have a counterpart will not affect the solution. To avoid having a hard threshold, we replace this step with a robust estimator using the Cauchy loss function, $\rho_\alpha(x) = \alpha^2 \ln(1 + \frac{x}{\alpha^2})$, where α is a parameter that controls where the loss begins to scale sublinearly. Similar to the approach in GICP, the robust estimator diminishes the effect of outliers while avoiding the removal of potential inliers. Consequently, our cost function becomes:

$$\mathsf{cost}_\rho(\mathbf{T}) \triangleq \sum_{k=1}^{n} \rho_\alpha(||\mathbf{x}_k^t - \mathbf{T} \cdot \mathbf{x}_k^s||^2_{\mathbf{C}_k}), \tag{6}$$

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

Following (2), the functions follows a representation such as $f_t(\cdot) = \sum_{j=1}^{m} \beta_j k_{\mathrm{SE}}(\cdot, \mathbf{z}_j)$. The Squared Exponential (SE) kernel has the form: $k_{\mathrm{SE}}(\mathbf{x}, \mathbf{z}) = \sigma_f^2 \exp(-\|\mathbf{x} - \mathbf{z}\|^2_{\mathbf{L}})$ where $\mathbf{L}$ is the diagonal matrix of *characteristic length-scales* and σ_f^2 is the signal variance. This is the most common kernel used in regression techniques using kernel methods [35] such as Gaussian processes [36] and Relevance Vector Machine (RVM) [37], and we choose it as part of the model selection due to its smoothness and being infinitely differentiable. Consequently, the regularizer term becomes

$$\mathsf{reg}(\mathbf{T}) = \lambda \sum_{k=1}^{n} (\sum_{j=1}^{m} \beta_j k_{\mathrm{SE}}(\mathbf{T} \cdot \mathbf{x}_k^s, \mathbf{z}_j) - f_s(\mathbf{x}_k^s))^2. \tag{7}$$

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

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

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

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

A. $\mathcal{H}_k$-Regularization via Sparse Bayesian Inference

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

$$y(\mathbf{x}; \mathbf{w}) \triangleq \sum_{j=1}^{n_b} w_j\phi_j(\mathbf{x}) = \mathbf{w}^\mathsf{T}\boldsymbol{\phi}(\mathbf{x}), \tag{8}$$

where $\boldsymbol{\phi}_i \triangleq \boldsymbol{\phi}(\mathbf{x}_i) = \mathrm{vec}(1, k_{\mathrm{SE}}(\mathbf{x}_1, \mathbf{x}_i) \ldots, k_{\mathrm{SE}}(\mathbf{x}_{n_b}, \mathbf{x}_i))$ are nonlinear basis functions. The weight vector, $\mathbf{w}$, is the model parameter whose distribution and dimension, n_b, to be learned [38], [39]. We note that the choice of this model is justified by the representation in (2).

The objective is to infer $\mathbf{w}$ such that $y(\mathbf{x}; \mathbf{w})$ generalizes well to new inputs $\mathbf{x}_*$ (test data). In this work, we use RVM [37] for the regression method. The sequential inference algorithm available for the RVM allows the method to be scalable while only a few (denoted by n_b here) basis functions with non-zero weights survive (*relevance vectors*) in the final model, resulting in a sparse model.

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

$$\mathbf{\Sigma} = (\mathbf{A} + \sigma^{-2}\mathbf{\Phi}^\mathsf{T}\mathbf{\Phi})^{-1}, \tag{9}$$

$$\boldsymbol{\mu} = \sigma^{-2}\mathbf{\Sigma}\mathbf{\Phi}^\mathsf{T}\mathbf{t}, \tag{10}$$

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

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

Algorithm 1 $\mathcal{H}_k$-Regularized Shape Registration

Require: Initial transformation $\mathbf{T}^{\mathsf{init}}$, target point cloud $\mathcal{X}_t$, source point cloud $\mathcal{X}_s$, optionally target function f_t;

1: $\mathbf{T}^{\mathsf{OPT}} \leftarrow \mathbf{T}^{\mathsf{init}}$ ▷ Initialize the transformation, e.g., $\mathbf{I}_4$
2: $f_s \leftarrow \mathtt{rvm\_train}(\mathcal{X}_s)$ ▷ Target values are corresponding intensity measurements.
3: **if** f_t not provided **then** ▷ In sequential data, the previous source function is the new target function.
4: $\quad f_t \leftarrow \mathtt{rvm\_train}(\mathcal{X}_t)$
5: **end if**
6: converged $\leftarrow$ false
7: **while** not converged **do**
8: $\quad \hat{\mathbf{T}} \leftarrow \mathbf{T}^{\mathsf{OPT}}$
9: $\quad \mathcal{I} \leftarrow \mathtt{nnsearch}(\mathcal{X}_s, \mathcal{X}_t, \hat{\mathbf{T}})$ ▷ Find Association using NN search
10: $\quad \mathbf{T}^{\mathsf{OPT}} \leftarrow \arg\min_{\mathbf{T} \in \mathrm{SE}(3)} \mathsf{cost}_\rho(\mathbf{T}) + \lambda\mathsf{reg}(\mathbf{T})$ ▷ Optimize over SE(3)
11: $\quad$ **if** $\mathrm{d}(\hat{\mathbf{T}}, \mathbf{T}^{\mathsf{OPT}}) < \epsilon$ **then** ▷ Check convergence using distance threshold ϵ
12: $\quad\quad$ converged $\leftarrow$ true
13: $\quad$ **end if**
14: **end while**
15: **return** $\mathbf{T}^{\mathsf{OPT}}$

B. Algorithmic Implementation

Algorithm 1 shows our implementation for solving the regularized form of the registration problem. In line 2 we learn the parameters of $f_s(\cdot)$ by maximizing the marginal likelihood, as presented in Subsection IV-A, following the

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

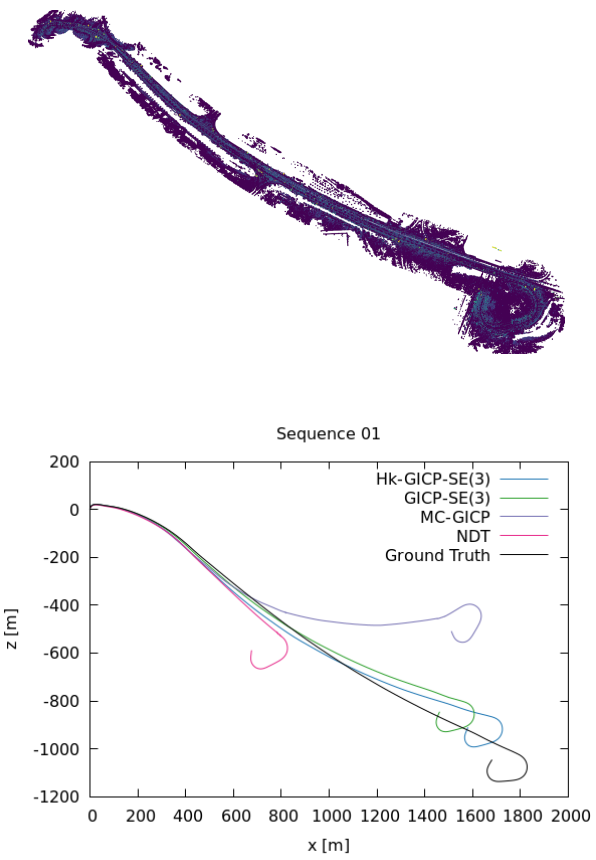

Fig. 2: Results of the proposed method versus the benchmark algorithms on sequences 01 of the KITTI Odometry dataset. Above are the point clouds projected into the same reference frame using the estimated $\mathcal{H}_k$-GICP-SE(3) odometry. This sequence is one of the more challenging ones, and we found that our proposed method had less transformation error by regularizing the cost on point cloud intensity.

sequential approach presented in [40]. The regularization term is added to the cost function in line 10. Finding the association is done using a NN search in line 9. The optimization is solved over $\mathrm{SE}(3)$ using the Conjugate-Gradient solver in the open source optimization library Ceres Solver [41].

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

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

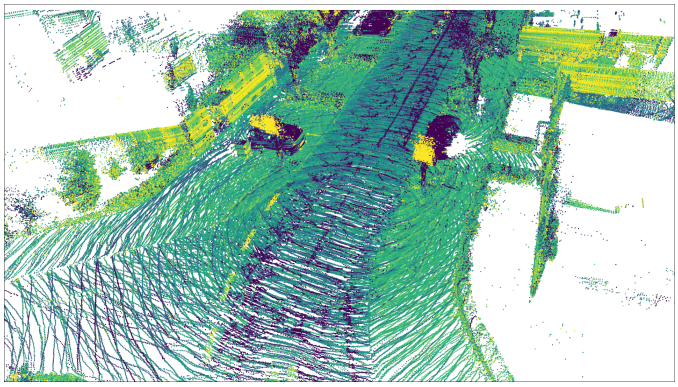

Fig. 3: A detailed view of sequence 00 reconstructed using $\mathcal{H}_k$-GICP-SE(3) odometry, labeled with intensity. Details such as lane lines and road signs are clearly visible, suggesting a good alignment.

V. Experimental Results

We now evaluate the proposed algorithm using LIDAR and RGB-D sensors. We use GICP-SE(3) [6], [29] as the baseline for comparison, since it is the algorithm we applied our regularizer too. We also compare to NDT [7], another method that does not use intensity or color, and Multichannel GICP (MC-GICP), a method that incorporates extra information into both the association and cost function [10]. For datasets where RGB data is available, we also compared to Color Supported GICP (GICP 6D) [9]. This method uses the distance in the CIELAB space to search for nearest neighbors. For NDT and GICP 6D we used the open source implementations available in the Point Cloud Library [42], while for MC-GICP we re-implemented the algorithm following the description in the paper. Our implementation of MC-GICP is available with the provided code for this paper, along with parameters used for each algorithm. In the first experiment, LIDAR data is from KITTI odometry dataset [43]. In the second experiment, we use RGB-D data from the TUM RGB-D SLAM dataset [44] to generate point clouds where the intensity values are computed using RGB measurements. Table I lists the parameters used for our algorithm.

A. LIDAR: KITTI Odometry dataset

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

1) Odometry Analysis: We first evaluated the proposed method versus our comparison methods in frame-to-frame odometry using the provided error metrics, an example is shown in Figure 2. Since all these methods use local gradient-based solvers, we seed the next frame with the solution of the previous, assuming there will not be a large change in velocity. The results for sequences 00 through 10 are presented in Table II. We found that the proposed method performed better overall, with a translation drift of 2.27% versus 2.66% for the GICP-SE(3), 3.08% for MC-GICP, and 5.07% for NDT. For the sequences that GICP-SE(3) had good results, the proposed

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

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

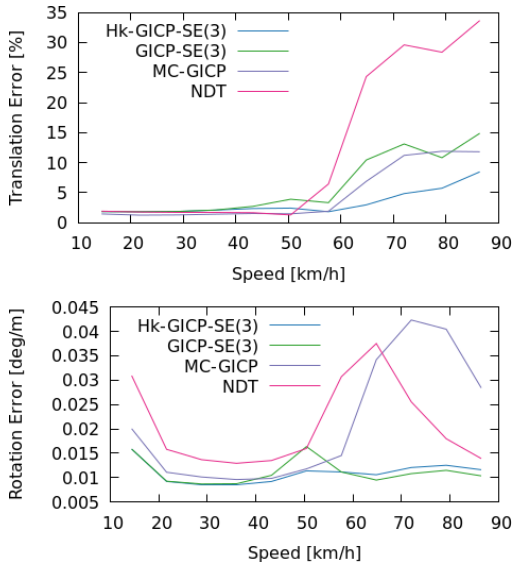

Fig. 4: Average translation and rotation error vs speed on the KITTI Odometry dataset. We can see from the translation error that $\mathcal{H}_k$-GICP-SE(3) has better results as the distance between point clouds increases.

$\mathcal{H}_k$-GICP-SE(3) performed similarly well without doing noticeably better. However, when GICP-SE(3) performed poorly, our method had noticeable improvement, suggesting the intensity regularizer contributes to minimizing the effect of poor geometrical registration (Figure 3). All methods showed good performance in terms of the rotation error results, even though the proposed method performed slightly better on average, 0.0160 °/m versus 0.0165 °/m for the GICP-SE(3) approach.

Figure 4 shows the average translation and rotation errors versus speed. We observed that the proposed method performs better at higher speeds, suggesting the intensity regularization aided to expand the basin of convergence of the GICP algorithm. In addition, the proposed approach is competitive with many of the methods on the KITTI odometry leader board. Most of those methods are SLAM or filtering systems that take into account observations from multiple frames for each position estimate. It is a good indication that our frame-to-frame approach is already competitive with these methods, and leaves open the possibility of incorporating this registration approach into a SLAM system for future work.

2) Convergence Analysis: We also used the KITTI Odometry dataset to evaluate the per-frame convergence of our proposed approach. To do so, we initialized the methods with the identity transformation and compared the initial error to the final error. The results of this analysis can be seen in Figure 5.

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

We also analyzed the computation time each algorithm takes, shown in Figure 6. Since our method includes training the functions online, it does take longer than the compared methods. It is approximately four times slower than GICP-SE(3) and seven times slower then NDT. There are compromises that can be made when constructing the regularizer, such as fewer RVM training iterations, that would make our method approach GICP-SE(3) in terms of speed and performance. There is also a potential for parallelization in training the sparse model as well as the cost function evaluation. Particularly if we change from the sequential approach presented in [40] to the batch solution first derived in [37]. However, such approaches would come at the cost of runtime when only a single thread is available for computation. The many independent but identical operations in batch-RVM training and cost function evaluation make implementing a version for the fine-grained parallelism of a GPU attractive as future work. None of the methods evaluated were quick enough to operate at the update rate of the LIDAR sensor used in this dataset (10 Hz). But the increased convergence performance of our approach suggest that it would work better when frames are dropped in an online system.

B. RGB-D: TUM RGB-D SLAM dataset

The TUM RGB-D SLAM dataset [44] was collected indoors using a Microsoft Kinect and a motion capture system for ground truth trajectory. We used data from four sequences: Freiburg 1 desk, Freiburg 2 desk, Freiburg 3 no-structure-texture-far, and Freiburg 3 no-structure-texture-near-with-loop. Parameters for all methods were tuned on Freiburg 1 xyz and Freiburg 1 rph. Depth images were associated with RGB images using the provided python program. We trained the regularizer functions on the intensity obtained by averaging the RGB channels of the images. The results presented in Table III show the per frame drift of the five methods using the provided

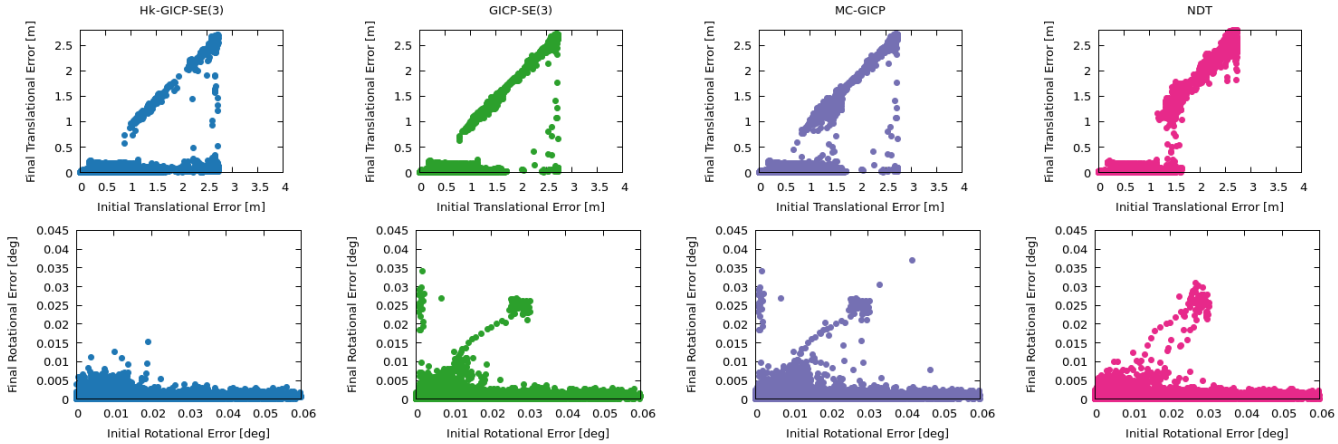

Fig. 5: Scatter plots of initial error versus final error of the various methods on KITTI Odometry sequence 00 through 10. Our approach more consistently converged to the ground truth transformation.

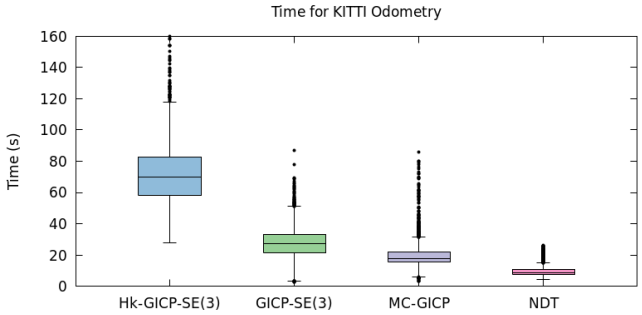

Fig. 6: Timing comparison for the various algorithms on the KITTI Odometry dataset for the 23900 consecutive pairs in all sequences, 00 through 10.

relative pose evaluation. Our approach does well in the two scenes with geometric structure (Freiburg 1 desk and Freiburg 2 desk) but show lower performance on the scenes that only have texture and no structure (both Freiburg 3 sequences). This presents the trade-off of the sparse Bayesian approach. The sparsity of support of the learned intensity functions, while suitable for convergence, performs poorly when there is no structure or local refinement. This is further illustrated in Figure 7 which provides sample images from one of the desk scenes and one of no structure scenes. It also includes CDF plots of translational error from the two desk scenes and from the two no structure scenes.

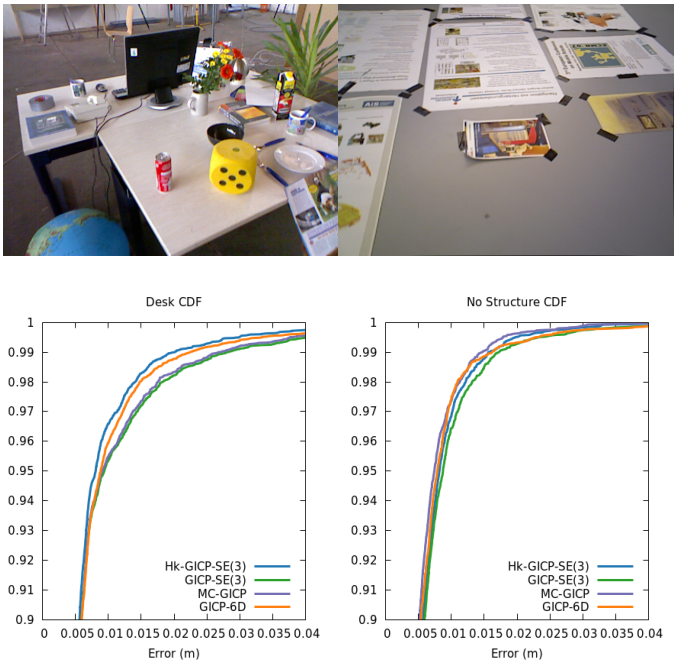

Fig. 7: Cumulative distribution plots for translational error from the TUM RGB-D SLAM dataset, comparing Freiburg 1 desk and Freiburg 2 desk to the Freiburg 3 no structure sequences.

VI. Conclusion

To reduce the effect of mis-associations in the registration problem, we presented an algorithmic approach to improve shape registration using regularizers represented in an RKHS. We presented results on real-world datasets using LIDAR and RGB-D sensors that showed promising improvements over transformation error when compared to related methods.

In the future, we could incorporate the proposed approach into a SLAM or smoothing framework, potentially a regularizer function could be trained on a local area, which could be then used to localize into and incrementally updated. We could also look at utilizing the probabilistic nature of the sparse Bayesian inference regularizer to make it more cohesive with the probabilistic motivation of GICP. And finally, regularizing to multi-dimensional information channels could be a useful direction to explore, particularly RGB [45].

Acknowledgment

This work was partially supported by the Toyota Research Institute (TRI), partly under award number N021515, however, this article solely reflects the opinions and conclusions of its authors and not TRI or any other Toyota entity.

References

[1] A. Makadia, A. Patterson, and K. Daniilidis, "Fully automatic registration of 3D point clouds," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, vol. 1. IEEE, 2006, pp. 1297–1304.

[2] G. Elbaz, T. Avraham, and A. Fischer, "3D point cloud registration for localization using a deep neural network auto-encoder," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2017, pp. 2472–2481.

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

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

[3] P. Besl and N. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, 1992.

[4] A. Censi, "An ICP variant using a point-to-line metric," in *Proc. IEEE Int. Conf. Robot. and Automation*. IEEE, 2008, pp. 19–25.

[5] Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," in *Proc. IEEE Int. Conf. Robot. and Automation*. IEEE, 1991, pp. 2724–2729.

[6] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. Syst. Conf.*, vol. 2, no. 4, 2009.

[7] T. Stoyanov, M. Magnusson, H. Andreasson, and A. J. Lilienthal, "Fast and accurate scan registration through minimization of the distance between compact 3D NDT representations," *Int. J. Robot. Res.*, vol. 31, no. 12, pp. 1377–1393, 2012.

[8] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3d data1," *Image and vision computing*, vol. 17, no. 2, pp. 135–147, 1999.

[9] M. Korn, M. Holzkothen, and J. Pauli, "Color supported generalized-ICP," in *International Conference on Computer Vision Theory and Applications*, vol. 3, Jan 2014, pp. 592–599.

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

[11] B. Huhle, M. Magnusson, W. Strasser, and A. J. Lilienthal, "Registration of colored 3d point clouds with a kernel-based extension to the normal distributions transform," in *Proc. IEEE Int. Conf. Robot. and Automation*, May 2008, pp. 4025–4030.

[12] J. Levinson, M. Montemerlo, and S. Thrun, "Map-based precision vehicle localization in urban environments." in *Proc. Robot.: Sci. Syst. Conf.*, vol. 4, 2007.

[13] J. Levinson and S. Thrun, "Robust vehicle localization in urban environments using probabilistic maps," in *Proc. IEEE Int. Conf. Robot. and Automation*, 2010, pp. 4372–4378.

[14] R. W. Wolcott and R. M. Eustice, "Visual localization within LIDAR maps for automated urban driving," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Chicago, IL, USA, September 2014, pp. 176–183.

[15] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for RGB-D cameras," in *Proc. IEEE Int. Conf. Robot. and Automation*, May 2013.

[16] J. Engel, J. Stueckler, and D. Cremers, "Large-scale direct SLAM with stereo cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, September 2015.

[17] B. K. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[18] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Int. Joint Conf. Artif. Intell.*, Vancouver, Canada, August 1981, pp. 674–679.

[19] C. Liu, J. Yuen, A. Torralba, J. Sivic, and W. T. Freeman, "SIFT flow: Dense correspondence across different scenes," in *Proc. European Conf. Comput. Vis.*, Marseille, France, October 2008, pp. 28–42.

[20] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics*, vol. 28, no. 3, p. 24, 2009.

[21] Y. Hu, R. Song, and Y. Li, "Efficient coarse-to-fine patchmatch for large displacement optical flow," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Las Vegas, NV, USA, June/July 2016, pp. 5704–5712.

[22] S. Vedula, S. Baker, P. Rander, R. Collins, and T. Kanade, "Three-dimensional scene flow," in *Proc. IEEE Int. Conf. Comput. Vis.*, Kerkyra, Greece, September 1999, pp. 722–729.

[23] M. Isard and J. MacCormick, "Dense motion and disparity estimation via loopy belief propagation," in *Proceedings of the Asian Conference on Computer Vision*, Hyderabad, India, 2006, pp. 32–41.

[24] D. Ferstl, G. Riegler, M. Ruether, and H. Bischof, "CP-Census: A novel model for dense variational scene flow from RGB-D data." in *Proc. British Mach. Vis. Conf.*, Nottingham, UK, September 2014.

[25] M. Hornacek, A. Fitzgibbon, and C. Rother, "SphereFlow: 6 DoF scene flow from RGB-D pairs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Columbus, OH, USA, 2014, pp. 3526–3533.

[26] M. Jaimez, M. Souiai, J. Gonzalez-Jimenez, and D. Cremers, "A primal-dual framework for real-time dense RGB-D scene flow," in *Proc. IEEE Int. Conf. Robot. and Automation*, Chicago, IL, USA, May 2015, pp. 98–104.

[27] Z. Yan and X. Xiang, "Scene flow estimation: A survey," *arXiv preprint arXiv:1612.02590*, 2016.

[28] A. Zaganidis, L. Sun, T. Duckett, and G. Cielniak, "Integrating deep semantic segmentation into 3-d point cloud registration," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2942–2949, oct 2018. [Online]. Available: https://doi.org/10.1109/lra.2018.2848308

[29] S. A. Parkison, L. Gan, M. Ghaffari Jadidi, and R. M. Eustice, "Semantic iterative closest point through expectation-maximization," in *Proceedings of the British Machine Vision Conference*, Newcastle, UK, September 2018, pp. 1–17.

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

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

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

[33] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2017.

[34] B. Schölkopf, R. Herbrich, and A. Smola, "A generalized representer theorem," in *Computational learning theory*. Springer, 2001, pp. 416–426.

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

[36] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT press, 2006, vol. 1.

[37] M. E. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. machine learning res.*, vol. 1, no. Jun, pp. 211–244, 2001.

[38] ——, "Bayesian inference: An introduction to principles and practice in machine learning," *Lecture notes in computer science*, vol. 3176, pp. 41–62, 2004.

[39] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[40] M. E. Tipping and A. C. Faul, "Fast marginal likelihood maximisation for sparse bayesian models." in *AISTATS*, 2003.

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

[42] R. B. Rusu and S. Cousins, "3D is here: Point Cloud Library (PCL)," in *Proc. IEEE Int. Conf. Robot. and Automation*, Shanghai, China, May 9-13 2011.

[43] A. Geiger, P. Lenz, and R. Urtasun, "are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012.

[44] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of RGB-D SLAM systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, Oct. 2012.

[45] M. Ghaffari, W. Clark, A. Bloch, R. M. Eustice, and J. W. Grizzle, "Continuous direct sparse visual odometry from RGB-D images," in *Proc. Robot.: Sci. Syst. Conf.*, Freiburg, Germany, June 2019.