2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

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

Ray Zhang, Tzu-Yuan Lin, Chien Erh Lin, Steven A. Parkison, William Clark, Jessy W. Grizzle, Ryan M. Eustice and Maani Ghaffari

***Abstract*— This paper 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.**

I. INTRODUCTION

Point cloud registration estimates the relative transformation between two noisy point clouds [1]–[5]. Point clouds obtained by RGB-D cameras, stereo cameras, and LIDARs contain rich color and intensity measurements besides the geometric information. The extra non-geometric information can improve the registration performance [6]–[8]. Deep learning can provide semantic attributes of the scene as measurements [9]–[11]. As illustrated in Fig. 1, this work focuses on the construction of a novel integrated framework to jointly process raw geometric and non-geometric information for point cloud registration.

Real-world applications such as SLAM [12] and 3D reconstruction [13] include noisy measurements, symmetries or dynamics objects, occlusion, and blurry observations. Examples are shown in Fig. 2. These cases make the data association process challenging. Existing Iterative Closest Point (ICP)-based work [1]–[3] approach this problem by adding appearance/semantic features [6], [7], [14], adding local or deep geometric features [3], [15], and introducing weighted many-to-many correspondences [16], [17].

*Toyota Research Institute (TRI) provided funds to support this work.

R. Zhang, T.Y. Lin, C.E. Lin, J. Grizzle, R. Eustice, and M. Ghaffari are with the University of Michigan, Ann Arbor, MI 48109, USA. {rzh,tzuyuan,chienerh,grizzle,eustice,maanigj}@umich.edu

S. Parkison is with TRI. steven.parkison@tri.global

W. Clark is with the department of Mathematics, Cornell University, Ithaca, NY. wac76@cornell.edu

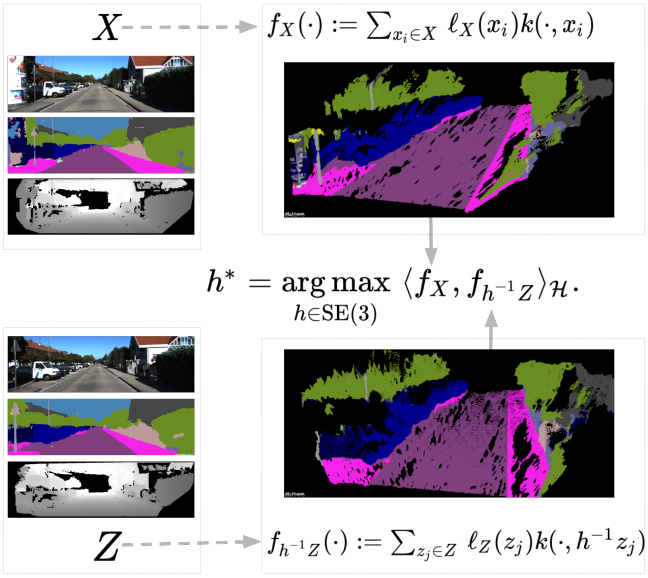

Fig. 1: Point clouds X and Z are represented by two continuous functions f_X, f_Z in a reproducing kernel Hilbert Space. Each point x_i has its own semantic labels, $\ell_X(x_i)$, encoded in the corresponding function representation via a tensor product representation. The registration is formulated as maximizing the inner product between two point cloud functions.

Gaussian Mixture Model (GMM) based registrations [18]–[20] 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 [19], [21]–[25], or by minimizing a distance measure between the two distribution and inferring the weight parameters [26]–[28].

This paper 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 [8], [14], 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 [29], [30]. 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

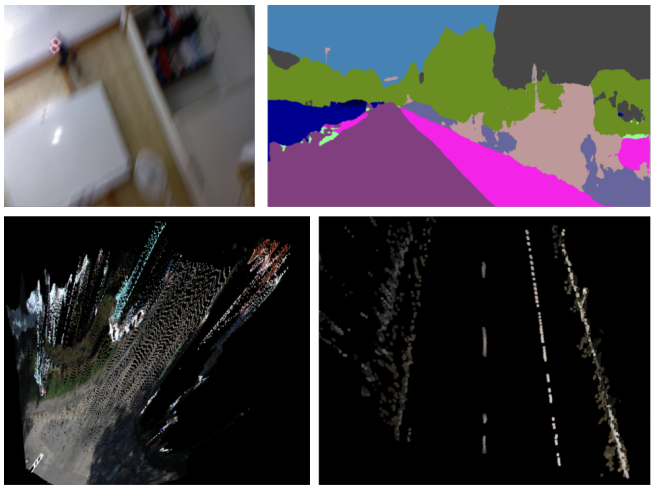

Fig. 2: Challenging scenes for stereo and RGB-D point cloud registration, including blurry image sources (from TUM RGB-D [32]), noisy semantic sources (from KITTI [33] and Nvidia [11]), noisy depth estimations (from KITTI), and highly repetitive patterns (from KITTI).

geometric and semantic information.

3) An open source GPU implementation available at [31]: https://github.com/UMich-CURLY/unified_cvo
4) Extensive evaluations using publicly available datasets for outdoor stereo and indoor RGB-D datasets.

II. Related Work

To improve the quality of one-to-one correspondences (*hard* assignment), the work of [34] assumes that only a portion of points can be paired thus only considers 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 [35] or from Gaussian weights [16]. EM-ICP [17] treats the correspondences as hidden variables, and use Expectation Maximization (EM) [36] to infer both the matches and then the transformations.

Point-to-plane [2], plane-to-plane [37], and Generalized-ICP [3] build local geometric structures to the loss formulation. The work of [38] combines multiple Euclidean invariant features. The work of [39] works on an IR camera and uses extra SIFT features from depth images to help keypoint correspondences. The work of [6] uses color/intensity for both association and registration. Color ICP [14] defines a sum of reprojected photometric and depth loss on dense RGB-D point clouds. GICP-RKHS [8] also appends an additional regularizer to the GICP's loss for point intensity via the Relevance Vector Machine [36]. Semantic-ICP [7] treats points' semantic labels and associations as additional hidden variables as a part of the EM-ICP framework. In our formulation, function representation combines both geometric and non-geometric information into a unified formulation, and it affects both the association and the optimization steps.

A. Mixture of Gaussian-based Registration Frameworks

Probabilistic registration frameworks represent point clouds as discrete [19], [40] or continuous probability densities [20]–[22], [25], [41], [42]. 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) [43].

Normal Distribution Transform (NDT) defines a discrete collection of bivariate Gaussian distributions to capture local surface structures [19], [44]. Discretization brings automatic soft data association without the need of inferring GMM weights. An effective discretization strategy requires suitable voxel sizes and efficient voxel deployment, for instance a forest of octrees [19], distance based voxel sizing [45], hierarchical voxel tree deployment [46], and cell clustering [47]. Comparing to NDT, the proposed method is also data association free, but it is further a continuous representation, thus avoids the above concerns caused by discretization.

Some continuous GMM-based methods minimize the distance between two distributions. Effective distance measures include Jensen-Shannon divergence [27] and the l_2 distance of the two dense [28] or sparse [41] GMMs. Kernel Correlation (KC) [26] maximizes the correlation between two point clouds using M-estimators, in particular a sum of Gaussian kernels. It has an identical loss function comparing to the proposed geometric only inner product, but its kernel length-scales 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.

B. Deep Learning in Registration

Fully connected layers with symmetric operations (max-pooling) in PointNet [48], [49], convolutions of sparse tensors in FCGF [15], sparse bilateral convolution in SPLATNet [50], and graph convolution layers in DGCNN [51] can capture local and global geometric features of point clouds. Examples of utilizing deep geometric features include PCR-Net [52], 3D-Feat-Net [53], PointNetLK [49].

Given extracted features, point correspondences are calculated in the many-to-many [54], [55] or one-to-one way [56], [57]. 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 [58]. DCP [55] 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 [59] 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.

This work is not an end-to-end deep learning solution, but our unified point cloud function representation can incorporate deep learning features, such as semantics, into the cost function. The potential way of using our inner product as a loss of an end-to-end framework can be a future study.

III. 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 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*

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

A. Constructing The Functions

We follow the same steps in [29] 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 $k : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}$:

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

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 to functions via

$$\begin{aligned} f_X(\cdot) &:= \sum_{x_i\in X} \ell_X(x_i)k(\cdot, x_i), \\ f_{h^{-1}Z}(\cdot) &:= \sum_{z_j\in Z} \ell_Z(z_j)k(\cdot, h^{-1}z_j). \end{aligned} \tag{3}$$

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

$$\langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}} := \sum_{x_i\in X, z_j\in Z} \langle \ell_X(x_i), \ell_Z(z_j)\rangle_{\mathcal{I}} \cdot k(x_i, h^{-1}z_j) \tag{4}$$

We use the kernel trick in machine learning [36], [60], [61] to substitute the inner products in (4) with the appearance kernel. After applying the kernel trick to (4), we get

$$\langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}} = \sum_{x_i\in X, z_j\in Z} k_c(\ell_X(x_i), \ell_Z(z_j))\cdot k(x_i, h^{-1}z_j), \tag{5}$$

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

B. 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

$$\begin{aligned} \langle \ell_X(x), \ell_Z(z)\rangle_{\mathcal{I}} &= \langle u_1 \otimes u_2 \otimes \dots, v_1 \otimes v_2 \otimes \dots\rangle \\ &= \langle u_1, v_1\rangle \cdot \langle u_2, v_2\rangle \cdot \dots. \end{aligned} \tag{6}$$

By substituting (6) into (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

$$\begin{aligned} \langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}} &= \sum_{x_i\in X, z_j\in Z} k(x_i, h^{-1}z_j) \cdot \prod_k k_{V_k}(u_{ki}, v_{kj}) \\ &:= \sum_{x_i\in X, z_j\in Z} k(x_i, h^{-1}z_j) \cdot c_{ij}. \end{aligned} \tag{7}$$

Equation (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 (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 (7) becomes

$$\langle f_X, f_{h^{-1}Z}\rangle_{\mathcal{H}} = \sum_{x_i\in X, z_j\in Z} k(x_i, h^{-1}z_j). \tag{8}$$

Through (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, (8) has the same formulation as Kernel Correlation [26].

C. An Indicator of Alignment

We want to have 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

$$\cos(\theta) = \frac{\langle f_X, f_Z\rangle_{\mathcal{H}}}{\|f_X\| \cdot \|f_Z\|}. \tag{9}$$

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 (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.*

$$i_\theta := \frac{1}{\sqrt{|X|\cdot|Z|}} \sum_{x_i\in X, z_j\in Z} c_{ij} \cdot k(x_i, z_j). \tag{10}$$

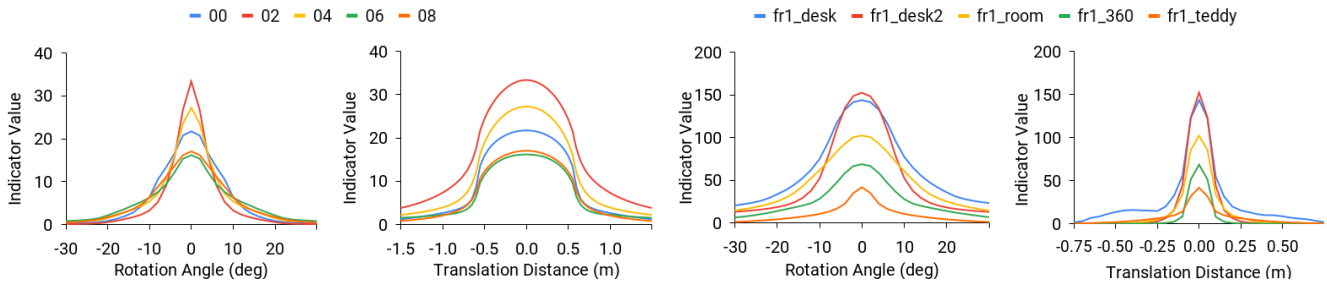

Fig. 3: Indicator value with respect to rotation angle and translation distance for KITTI Stereo (left figures) and TUM RGB-D (right figures) sequences.

The behavior of the alignment indicator with respect to the rotation and translation errors is shown in Fig. 3. 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 [62] uses a neural network to predict a similar metric and detect loop closures. The cosine of the angle in* (9) *or the indicator in* (10) *provide such a metric for* self-supervised *learning while taking into account the semantic information. Given the promising results of [62], the combination of our metric with deep learning is an interesting future research direction.*

D. Optimization

We perform gradient ascent over $SE(3)$ on the inner product in (7). The flow and step size expressions are in the same forms as [29]. During the iterative optimization, the alignment indicator in III-C can guide the lengthscale update. When the lengthscale is large, each point is associated with farther points, which provides the point cloud function 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.

IV. 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 is 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 previous 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 [63] implemented in OpenCV [64]. 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.

V. Experimental Results

We now present the frame to frame registration experiments on both outdoor and indoor datasets: KITTI stereo odometry [65] and TUM RGB-D dataset [32].

A. Experimental Setup

All 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 [31]. The same values were used within one dataset.

On KITTI, our baselines are GICP [3], Multichannel-ICP [6], and 3D-NDT [19]. 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 [66]. The Multichannel-ICP implementation is from [8]. Both the baselines and the proposed methods remove the first 100 rows of image pixels that mainly include sky pixels, as well as 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. VI

On TUM RGB-D, we use the same baselines for geometric registration as KITTI. We compare Color CVO with Dense Visual Odometry (DVO) [67] and Color ICP [14]. We reproduced DVO results with the code from [68] because the original DVO source code requires an outdated ROS dependency [69]. The Color ICP implementation is taken from Open3D [70]. The baselines use full point clouds.

B. Outdoor Stereo Camera: KITTI Stereo Odometry

We select a subset of pixels from KITTI's stereo images via OpenCV's FAST [63] feature detector. The depth values of the selected pixels are generated with ELAS [72]. The semantic predictions of the images come from Nvidia's pre-trained neural network [11], which was trained on 200 labeled images. Examples of the point clouds are in Fig. 2. 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 I. From sequence 00 to

TABLE I: Results of the proposed frame-to-frame method using the KITTI [33] stereo odometry benchmark as evaluated on the average drift in translation, as a percentage (%), and rotation, in degrees per meter($^\circ$/m). The drifts are calculated for all possible subsequences of $100, 200 \ldots, 800$ 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 ($^\circ$/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 [3]	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 ($^\circ$/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 [19]	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 ($^\circ$/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 ($^\circ$/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 [6]	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 ($^\circ$/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 (with color)	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**
	r ($^\circ$/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**

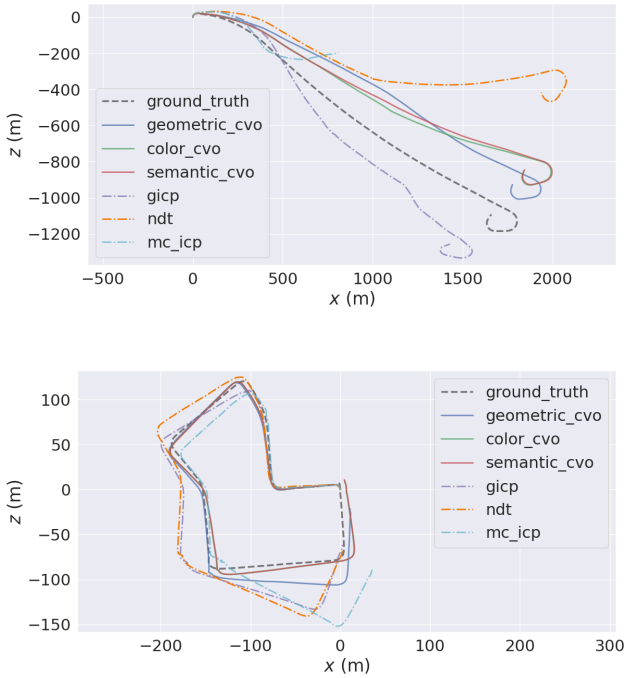

Fig. 4: An illustration of the proposed registration method on KITTI stereo sequence 01 (top) and 07 (bottom) versus the baselines. The black dashed trajectory is the ground truth. The dot-dashed trajectories are the baselines. Plotted with EVO [71].

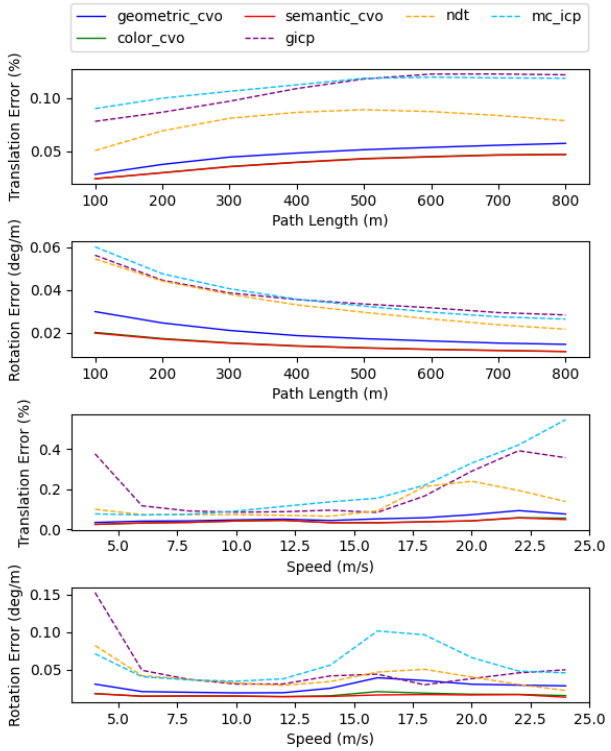

Fig. 5: From top to down: the average translation errors and rotation errors on KITTI Stereo sequences 00 to 10 with respect to the distance segment and the moving speed, respectively.

10, our geometric method has a lower average translational error (4.55%) comparing 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%). If we add semantic information the error is further reduced (3.64%). The addition of color and semantic information also yields a lower standard deviation. Meanwhile, the average rotational drift of the proposed methods are 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 Fig. 4). Figure 5 shows the average translational and rotational errors at different distances and speeds. The proposed methods show 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 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 sec, 6.6 sec, and 57 sec per frame on CPU, respectively.

C. 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 [63] feature selector.

TABLE II: 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 [3]		3D-NDT [19]		Color CVO		DVO [67]		Color ICP [14]	
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

TABLE III: 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 $\mathrm{deg/sec}$.

			Geometric CVO		GICP [3]		3D-NDT [19]		Color CVO		DVO [67]		Color ICP [14]	
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**

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

VI. Discussions and Limitations

Besides the reported baseline results, we tried to run Semantic ICP [7] and Color ICP [14] with KITTI's full stereo point clouds as well. However, Semantic-ICP takes $4-8\,\mathrm{min}$ per frame on our machine, thus is infeasible to complete all the 23190 frames. The original Color ICP work has not been tuned for stereo data, and failed on KITTI sequence 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 [73] was unable to complete some challenging sequences such as KITTI sequence 01. We cannot use PCL's Voxel Filter [66] either because the original color and semantic information is lost during its downsampling. Only FAST [63] feature selector from OpenCV [64] works for all the tested datasets. A future direction is to find a more robust downsampling scheme for this framework.

Moreover, the proposed methods' performance relies on the geometric lengthscale during the optimization. Adaptive CVO [74] addresses the lengthscale decay by regarding it as a part of the optimizing variable. Still we need to manually choose an initial lengthscale. For inputs with larger accelerations, the lengthscale needs a global perspective for such abrupt changes. In this case, another future direction is an algorithmic way of selecting the initial lengthscale, or more broadly, studying the hyperparameter learning problem.

VII. Conclusion

We developed a nonparametric registration framework that integrates geometric and semantic measurements and does not require explicit data association. The proposed approach can utilize the extra visual and semantic information from modern range sensors while not being restricted by pairwise data correspondences. The novel hierarchical distributed representation of features via the tensor product representation provides a mathematically sound and systematic way of incorporating semantic knowledge into the point cloud model. 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. We also provided an open-source GPU implementation.

In the future, we shall explore the connections of the developed framework with deep learning for representation learning in applications such as multi-modal feature learning, place recognition, and robust tracking and SLAM in harsh and visually degraded situations.

REFERENCES

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

[2] Y. Chen and G. G. Medioni, "Object modeling by registration of multiple range images." *Image Vision Comput.*, vol. 10, no. 3, pp. 145–155, 1992.

[3] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Robotics: science and systems*, vol. 2, no. 4. Seattle, WA, 2009, p. 435.

[4] F. Pomerleau, F. Colas, and R. Siegwart, *A Review of Point Cloud Registration Algorithms for Mobile Robotics*, 2015.

[5] L. Cheng, S. Chen, X. Liu, H. Xu, Y. Wu, M. Li, and Y. Chen, "Registration of laser scanning point clouds: A review," *Sensors*, vol. 18, no. 5, p. 1641, 2018.

[6] J. Servos and S. L. Waslander, "Multi channel generalized-ICP," in *Proc. IEEE Int. Conf. Robot. and Automation*. IEEE, 2014, pp. 3644–3649.

[7] S. A. Parkison, L. Gan, M. G. Jadidi, and R. M. Eustice, "Semantic iterative closest point through expectation-maximization." in *Proc. British Mach. Vis. Conf.*, 2018, p. 280.

[8] S. A. Parkison, M. Ghaffari, L. Gan, R. Zhang, A. K. Ushani, and R. M. Eustice, "Boosting shape registration algorithms via reproducing kernel Hilbert space regularizers," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 4563–4570, 2019.

[9] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 3431–3440.

[10] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2017.

[11] Y. Zhu, K. Sapra, F. A. Reda, K. J. Shih, S. Newsam, A. Tao, and B. Catanzaro, "Improving semantic segmentation via video propagation and label relaxation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 8856–8865.

[12] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "Elasticfusion: Real-time dense slam and light source estimation," *Int. J. Robot. Res.*, vol. 35, no. 14, pp. 1697–1716, 2016.

[13] M. Zollhöfer, P. Stotko, A. Görlitz, C. Theobalt, M. Nießner, R. Klein, and A. Kolb, "State of the art on 3d reconstruction with rgb-d cameras," in *Computer graphics forum*, vol. 37, no. 2. Wiley Online Library, 2018, pp. 625–652.

[14] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 143–152.

[15] C. Choy, J. Park, and V. Koltun, "Fully convolutional geometric features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 8958–8966.

[16] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2d and 3d point matching: Pose estimation and correspondence," *Pattern recognition*, vol. 31, no. 8, pp. 1019–1031, 1998.

[17] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *Proc. European Conf. Comput. Vis.*, 2002, pp. 418–432.

[18] P. Biber and W. Strasser, "The normal distributions transform: a new approach to laser scan matching," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.*, vol. 3, Oct 2003, pp. 2743–2748 vol.3.

[19] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, 2007.

[20] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug 2011.

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

[22] R. Horaud, F. Forbes, M. Yguel, G. Dewaele, and J. Zhang, "Rigid and articulated point registration with expectation conditional maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 587–602, 2010.

[23] B. Eckart, K. Kim, A. Troccoli, A. Kelly, and J. Kautz, "Mlmd: Maximum likelihood mixture decoupling for fast and accurate point cloud registration," in *2015 International Conference on 3D Vision*. IEEE, 2015, pp. 241–249.

[24] B. Eckart and A. 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*. IEEE, 2013, pp. 4355–4362.

[25] B. Eckart, K. Kim, and J. Kautz, "Hgmr: Hierarchical Gaussian mixtures for adaptive 3D registration," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 705–721.

[26] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *European conference on computer vision*. Springer, 2004, pp. 558–569.

[27] F. Wang, B. Vemuri, and A. Rangarajan, "Groupwise point pattern registration using a novel cdf-based jensen-shannon divergence," *Proceedings / CVPR, IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 1283–1288, 07 2006.

[28] Bing Jian and B. C. Vemuri, "A robust algorithm for point set registration using mixture of Gaussians," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1246–1251.

[29] 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.

[30] W. Clark, M. Ghaffari, and A. Bloch, "Nonparametric continuous sensor registration," 2020.

[31] UMich-CURLY. (2020) A new framework for registration of semantic point clouds from stereo and rgb-d cameras. [Online]. Available: https://github.com/UMich-CURLY/unified_cvo

[32] 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.

[33] 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.

[34] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299 – 309, 2005.

[35] A. Rangarajan, H. Chui, and J. S. Duncan, "Rigid point feature registration using mutual information," *Medical Image Analysis*, vol. 3, no. 4, pp. 425–440, 1999.

[36] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 2006.

[37] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, ser. SGP 04. New York, NY, USA: Association for Computing Machinery, 2004, p. 2231.

[38] G. C. Sharp, S. W. Lee, and D. K. Wehe, "ICP registration using invariant features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, pp. 90–102, 2002.

[39] A. Sehgal, D. Cernea, and M. Makaveeva, "Real-time scale invariant 3d range point cloud registration," 06 2010, pp. 220–229.

[40] P. Biber and W. StraSSer, "The normal distributions transform: A new approach to laser scan matching," vol. 3, 11 2003, pp. 2743 – 2748 vol.3.

[41] D. Campbell and L. Petersson, "An adaptive data representation for robust point-set registration and merging," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4292–4300.

[42] G. D. Evangelidis and R. Horaud, "Joint alignment of multiple point sets with batch and incremental expectation-maximization," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 6, pp. 1397–1410, 2017.

[43] A. Berlinet and C. Thomas-Agnan, *Reproducing Kernel Hilbert Space in Probability and Statistics*, 01 2004.

[44] P. Biber, S. Fleck, and W. Straßer, "A probabilistic framework for robust and accurate matching of point clouds," in *Joint Pattern Recognition Symposium*. Springer, 2004, pp. 480–487.

[45] Y. Miao, Y. Liu, H. Ma, and H. Jin, "The pose estimation of mobile robot based on improved point cloud registration," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 52, 2016.

[46] C. Ulaş and H. Temeltaş, "3D multi-layered normal distribution transform for fast and long range scan matching," *Journal of Intelligent & Robotic Systems*, vol. 71, no. 1, pp. 85–108, 2013.

[47] A. Das and S. L. Waslander, "Scan registration using segmented region growing NDT," *Int. J. Robot. Res.*, vol. 33, no. 13, pp. 1645–1663, 2014.

[48] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2017, pp. 652–660.

[49] Y. Aoki, H. Goforth, R. A. Srivatsan, and S. Lucey, "PointNetLK: Robust & efficient point cloud registration using pointnet," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, June 2019.

[50] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. Kautz, "SPLATNet: Sparse lattice networks for point cloud processing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2530–2539.

[51] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, 2019.

[52] V. Sarode, X. Li, H. Goforth, Y. Aoki, R. A. Srivatsan, S. Lucey, and H. Choset, "Pcrnet: Point cloud registration network using pointnet encoding," *ArXiv*, vol. abs/1908.07906, 2019.

[53] Z. J. Yew and G. H. Lee, "3dfeat-net: Weakly supervised local 3d features for point cloud registration," in *Proc. European Conf. Comput. Vis.* Springer, 2018, pp. 630–646.

[54] 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.*, 2017, pp. 4631–4640.

[55] Y. Wang and J. M. Solomon, "Deep closest point: Learning representations for point cloud registration," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3523–3532.

[56] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016.

[57] 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 *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 7191–7201.

[58] W. Lu, G. Wan, Y. Zhou, X. Fu, P. Yuan, and S. Song, "Deepvcp: An end-to-end deep neural network for point cloud registration," *Proc. IEEE Int. Conf. Comput. Vis.*, pp. 12–21, 2019.

[59] C. Choy, W. Dong, and V. Koltun, "Deep global registration," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

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

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

[62] X. Chen, T. Läbe, A. Milioto, T. Röhling, O. Vysotska, A. Haag, J. Behley, C. Stachniss, and F. Fraunhofer, "OverlapNet: Loop closing for LiDAR-based SLAM," in *Proc. Robot.: Sci. Syst. Conf.*, 2020.

[63] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conf. Comput. Vis.* Springer, 2006, pp. 430–443.

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

[65] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.

[66] 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.

[67] C. Kerl, J. Sturm, and D. Cremers, "Dense visual slam for RGB-D cameras," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots and Syst.* IEEE, 2013, pp. 2100–2106.

[68] M. Pizenberg, "DVO (without ROS dependency)," https://github.com/mpizenberg/dvo/tree/76f65f0c9b438675997f595471d39863901556a9, 2019.

[69] C. Kerl, "Dense Visual Odometry (dvo)," https://github.com/tum-vision/dvo, 2013.

[70] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," *arXiv:1801.09847*, 2018.

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

[72] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *Proc. Asian Conf. Comput. Vis.*, 2010.

[73] J. Engel, V. Koltun, and D. Cremers, "Direct sparse odometry," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 3, pp. 611–625, 2017.

[74] T.-Y. Lin, W. Clark, R. M. Eustice, J. W. Grizzle, A. Bloch, and M. Ghaffari, "Adaptive continuous visual odometry from RGB-D images," *arXiv preprint arXiv:1910.00713*, 2019.