Deep Hough Voting for Robust Global Registration

Junha Lee   Seungwook Kim   Minsu Cho   Jaesik Park

POSTECH CSE & GSAI

http://cvlab.postech.ac.kr/research/DHVR

Abstract

Point cloud registration is the task of estimating the rigid transformation that aligns a pair of point cloud fragments. We present an efficient and robust framework for pairwise registration of real-world 3D scans, leveraging Hough voting in the 6D transformation parameter space. First, deep geometric features are extracted from a point cloud pair to compute putative correspondences. We then construct a set of triplets of correspondences to cast votes on the 6D Hough space, representing the transformation parameters in sparse tensors. Next, a fully convolutional refinement module is applied to refine the noisy votes. Finally, we identify the consensus among the correspondences from the Hough space, which we use to predict our final transformation parameters. Our method outperforms state-of-the-art methods on 3DMatch and 3DLoMatch benchmarks while achieving comparable performance on KITTI odometry dataset. We further demonstrate the generalizability of our approach by setting a new state-of-the-art on ICL-NUIM dataset, where we integrate our module into a multi-way registration pipeline.

1. Introduction

Point cloud registration is one of the fundamental tasks in modern 3D computer vision, aiming to estimate the rigid transformation parameter to align a partially overlapping pair of point cloud fragments. It plays a vital role in autonomous driving, robotics, augmented reality, and other various applications. The objective function of pairwise point cloud registration can be formulated as:

$$E(\mathbf{R}, \mathbf{t}) = \frac{1}{N} \sum_{(\mathbf{p}, \mathbf{q}) \in \mathcal{C}} \|\mathbf{q} - \mathbf{R}\mathbf{p} - \mathbf{t}\|^2, \qquad (1)$$

where $\mathbf{p}, \mathbf{q} \in \mathbb{R}^3$ are points of a correspondence, $\mathcal{C}$ is the set of all true correspondences between the point cloud pair, N is the cardinality of $\mathcal{C}$, and $\mathbf{R} \in SO(3), \mathbf{t} \in \mathbb{R}^3$ are the rotation matrix and the translation vector, respectively. Therefore, point cloud registration is a dual problem of
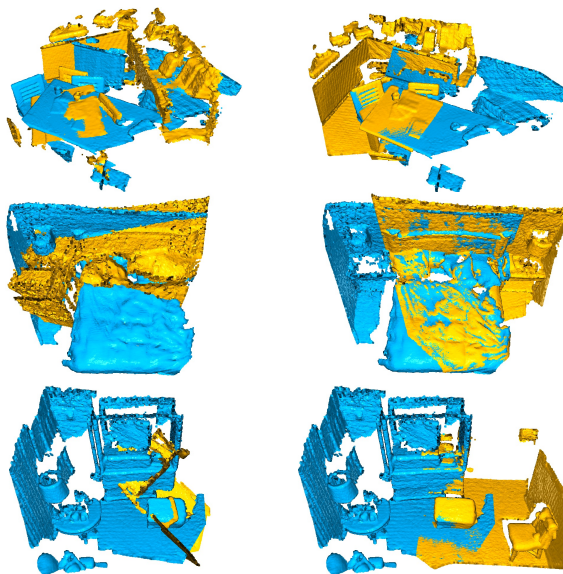

Figure 1. Pairwise point cloud registration results of DGR [10] (first column) and ours (second column) on 3DMatch dataset [56]. Our method is robust under low overlap and repetitive structures while being about twice as fast as DGR. Best viewed in color.

finding inlier correspondences and optimizing transformation parameters to minimize the geometric errors on these correspondences. Prior research on this topic can be categorized into geometric feature descriptors [4, 13, 16, 17, 23, 44, 50], pose optimization algorithms [5, 19, 58], and the recent learning-based feature extraction and registration pipelines [2, 10, 22, 52, 53]. Despite the impressive performances of recent end-to-end pipelines, they either require a strong assumption on the high overlapping ratio of point clouds [52], lack scalability to be operable on large-scale point clouds [53], or rely on iterative post-processing [10].

We propose a novel robust registration pipeline for large-scale point cloud fragment pairs to address these issues, yielding accurate rotation and translation predictions even in low point cloud overlap ratios. The key idea is to view the registration problem as finding a consensus among the candidate correspondences on a *discretized* 6D transforma-
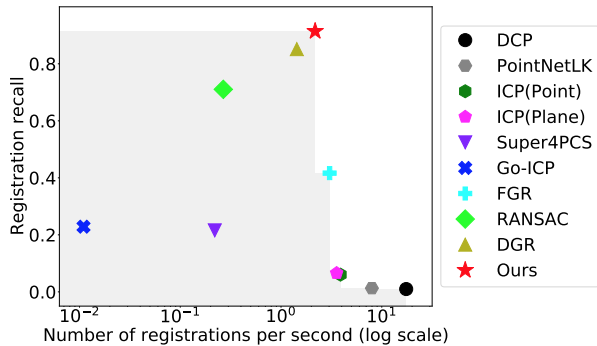
Figure 2. Registration recall and the number of registration per second in log scale on 3DMatch benchmark. Our approach is the most accurate overall and fastest among learning-based methods.

tion space, leveraging Hough voting with sparse tensors. Using the noisy candidate correspondences, we accumulate votes into discrete bins of the Hough space. After all the votes are cast, we use a learnable refinement module on the Hough space to filter the noisy vote values. Finally, the bin with maximum votes implies consensus among the candidate correspondences, which we use as our final prediction for the transformation parameters between the point cloud pair. The Hough space is sparsely constructed to facilitate small-sized bins for higher accuracy while incurring significantly less memory overhead than a dense parameter space.

The contributions of our work are as follows:

- We propose a novel approach for point cloud registration using Hough voting, where our *sparse* 6D parameter space improves the accuracy and memory efficiency of our approach,
- We propose a learnable refinement module on the Hough space to enhance the representability of our sparse parameter space for increased robustness,
- We achieve a new state-of-the-art on the real-world indoor 3DMatch [56]/3DLoMatch [30] datasets, and perform comparably on the outdoor KITTI [21] dataset,
- We prove our approach's generalizability by integrating our into a multi-way registration pipeline, evaluated on [26] to set a new state of the art,
- Our method exhibits faster speed compared to other learning-based registration pipelines.

2. Related Work

There are numerous approaches to solve point cloud registration. Prior work tackles the registration problem by decoupling the problem into a geometric feature description and robust matching problem.

Learning-based local feature. PointNet [42] uses shared MLP network to encode individual coordinates of the input point cloud into high dimensional embedding space, and performs a permutation-invariant operation to aggregate features into a single global descriptor. PerfectMatch [23] uses a smoothed density value (SDV) representation computed for each interest point, demonstrating rotation invariance and high generalizability. FCGF [13] leverages a fully-convolutional 3D network and metric learning losses to extract features that outperform PerfectMatch while being up to 290 times faster. However, these approaches focus mainly on feature descriptors, with little attention to feature detectors. D3Feat [4] therefore predicts detection scores with the features for each 3D point, showing that sampling features with higher scores is sufficient to achieve accurate and fast alignment. These learning-based approaches outperform hand-crafted geometric features [31, 45, 47, 48, 51] which are sensitive to hyperparameter settings.

Robust model fitting. Correspondences obtained from feature matching often contain high proportions of outliers, which have to be filtered out for robust point cloud registration. One of the most widely used methods is RANdom SAmple Consensus (RANSAC) and its variants [1, 27, 37, 45, 49], where a set of candidate correspondences are sampled every iteration, and an alignment is produced based on these correspondences to be evaluated.

Hough transform [29] was first introduced as a voting scheme to detect lines by discretizing parameter space into bins. SIFT [34] clusters features in pose space using Hough transform to estimate object poses. PHM [8] and HPF [39] applies this idea to efficient and robust matching algorithms for the task of object discovery and semantic correspondence, respectively. VoteNet [41] votes for object centers in 3D space to predict 3D bounding boxes for object detection. More recently, CHMNet [38] proposes to carry out local geometric matching using the idea of Hough voting to identify matches between images of the same category.

Our approach is inspired by recent methods using Hough voting, and we discretize the spatial transformation parameter space into sparse bins. We then vote on the sparse 6D space to predict the transformation parameters robustly based on the consensus from feature correspondences.

Global registration frameworks. Iterative Closest Points (ICP) [5] is a widely used approach due to its reliable performance in local registration. However, it requires an appropriate initial pose to avoid local minima and performs expensive nearest-neighbor queries in their inner loops. Other approaches make use of branch-and-bound [36], semi-definite programming [35], or maximal clique selection [54]. These methods exhibit high accuracy but require time-consuming iterative sampling or heavier computation as the inlier ratio decreases. FGR [58] and SparseICP [6] use robust objective functions to filter out outliers during optimization. While FGR shows much shorter registration

times albeit comparable performances, it is sensitive to the configurations of hand-crafted feature [45].

End-to-end registration frameworks perform the task of feature learning and pose optimization jointly in a single forward step. PointNetLK [2] combines PointNet [42] global features with an iterative pose optimization method into a single recurrent deep neural network. Deep Closest Points (DCP) [52] addresses local optima issues and other difficulties in traditional ICP [5], but it has a strong assumption that the correspondences are distributed over the entire point set. While PRNet [53] integrates a keypoint selection step to DCP, making it applicable to aligning point cloud pairs with partial overlap, it is not feasible to be applied to large-scale point sets due to computational complexity. Gojcic *et al.* [22] utilizes FCGF, incorporating a set filtering network [57] and extending to multi-way registration problem. Deep Global Registration [10] utilizes a 6D sparse convolutional network [12] to predict the inlier probabilities of feature correspondences, which are then used as weights to predict the rigid transformation parameters. However, it requires an iterative post-processing step for refinement. PointDSC [3] explicitly incorporates spatial consistency for pruning outlier correspondences prior to registration. Predator [30] proposes the overlap-attention block for early information exchange between the latent encodings of the point cloud pair, being effective even under very low overlap ratio.

Our proposed method integrates a learning-based feature extractor together with the highly robust sparse Hough voting module and a single-shot refinement module. Despite being simpler, without any iterative processing steps or intensive outlier rejection methods, our proposed method shows improved efficacy over prior methods on real-world large-scale 3D scans.

3. Method

The proposed point cloud registration network is comprised of three steps: (1) local feature extraction, (2) sparse Hough voting, and (3) Hough space refinement. Each step is detailed in the following subsections.

3.1. Local Feature Extraction

Given a point cloud pair, we extract local features that represent the geometric context of each point in metric feature space. These features are used to compute putative correspondences between the point clouds using nearest neighbor search. For each point in a point cloud, the top-1 nearest neighbor is chosen as a candidate correspondence. Therefore, given a point cloud pair with $\mathcal{M}$ and $\mathcal{N}$ points, respectively, we can obtain up to $\mathcal{M} + \mathcal{N}$ unique candidate correspondences in this step. Our work adopts Fully Convolutional Geometric Features (FCGF) [13] as the local feature extractor. Not only do features of FCGF show superior

matching accuracy, but they also have low dimensionality of around 16 to 32, which facilitates the quick neighbor search.

Next, we construct a set of triplets for the subsequent Hough voting step, where each triplet consists of three unique candidate correspondences between the point cloud pair. Motivated by FGR [58], we perform simple tuple tests to filter out triplets with obviously spurious correspondences. In our work, considering a triplet with correspondences $(\mathbf{p_1}, \mathbf{q_1})$, $(\mathbf{p_2}, \mathbf{q_2})$ and $(\mathbf{p_3}, \mathbf{q_3})$, the tuple test checks the following point-to-point distance conditions:

$$\forall i \neq j, | \, \|\mathbf{p}_i - \mathbf{p}_j\| - \|\mathbf{q}_i - \mathbf{q}_j\| \, | < 3 \times v, \qquad (2)$$

where v is the spatial hashing size, or the voxel size. The distance threshold $3 \times v$ is chosen as it was speculated to be the approximate upper bound to the error caused by spatial hashing. This tuple test is effective under the invariance condition that rotation and translation transformations do not alter the point-to-point distances in a point cloud. While many other methods propose to filter out outliers more aggressively either by using iterative algorithms [14, 19, 20, 28, 43] or learning-based methods [7, 12, 40, 57], we propose that our registration pipeline is highly robust to outliers, depreciating the necessity of additional, extensive outlier filtering. From here onwards, we use $\mathcal{C}$ to denote the set of triplets that passes Eq. (2).

3.2. Sparse Hough Voting

We adapt Hough voting to discover the most sound consensus between the candidate correspondences to predict spatial transformation parameters. In our context, the spatial transformation parameters to predict are rotation and translation parameters in 3D space. We use the axis-angle representation [55] to represent rotation in 3D space and three parameters along x-, y-, and z-axes for translation, having six degrees of freedom in total. Therefore, we construct a discretized 6D parameter space, where each discrete unit is called a *bin*. Our objective is to identify the bin representing the transformation parameters that best represent the consensus among the candidate correspondences to be used as our final transformation parameter prediction.

Using axis-angle representation, we can represent a rotation as a 3D vector $\mathbf{r} \in \mathbb{R}^3$, where $\mathbf{r}/\|\mathbf{r}\|$ is the rotation axis and $\|\mathbf{r}\|$ is the rotation angle. We can convert a given rotation matrix $\mathbf{R}$ to its axis-angle representation $\mathbf{r}$ as,

$$\|\mathbf{r}\| = \arccos\left(\frac{\text{Tr}(\mathbf{R}) - 1}{2}\right),$$
$$\frac{\mathbf{r}}{\|\mathbf{r}\|} = \frac{1}{2 \sin \theta} \begin{bmatrix} \mathbf{R}_{32} - \mathbf{R}_{23} \\ \mathbf{R}_{13} - \mathbf{R}_{31} \\ \mathbf{R}_{21} - \mathbf{R}_{12} \end{bmatrix}, \qquad (3)$$

where $\mathbf{R}_{ij}$ is the element in $\mathbf{R}$ in i-th row and j-th column, $\text{Tr}(\cdot)$ is the matrix trace operator. An axis-angle representa-
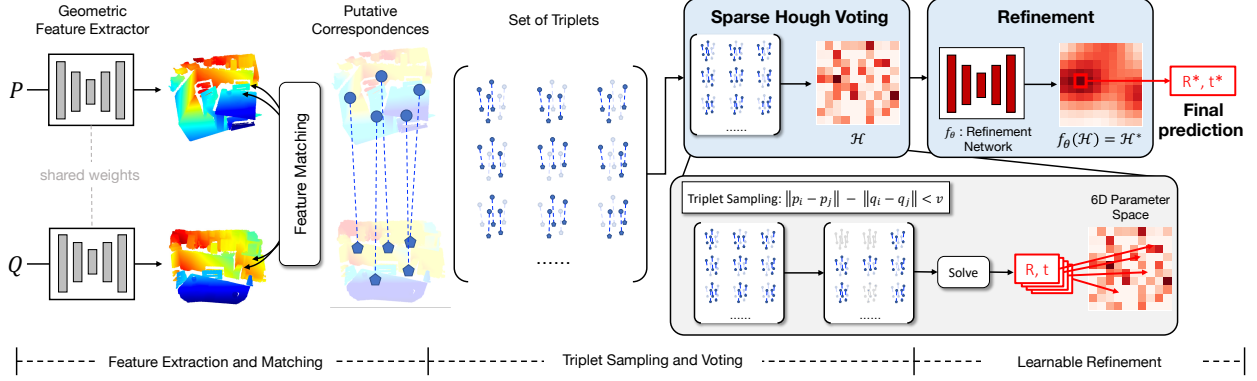
Figure 3. Illustration of our proposed module. We compute feature correspondences between point cloud pairs from locally extracted features, which are used to construct a set of triplets. Each triplet votes on the corresponding bin in the sparse Hough space in a parallelized fashion. We then perform a fully convolutional Hough space refinement to suppress the noisy Hough voting space. Finally, we identify the bin with the max votes, from where we make our final predictions on the transformation parameters.

tion of 3D rotation is in the form of a sphere, while translation parameters can simply be represented in a cuboid space of x-, y-, and z-axes translations. Together, the combined Hough space for rotation and translation would typically require a 6D space. However, a dense and high-dimensional voting space is redundant, and translation parameters are unbounded in space, posing technical difficulties.

To this end, we use a *sparse* 6D Hough space, leveraging the *sparse tensor representation* provided by [11]. Our implementation of sparse Hough space incurs minimal memory costs with no redundancy, as storage is required only for bins of the Hough space where votes are cast on. This eschews the issue of translation parameters being unbounded as well. Our sparse 6D Hough space $\mathcal{H}$ can represent all possible rotations and translations between two point clouds, where we can accumulate votes from the set of triplets onto the sparse bins. For each triplet, we need to retrieve the optimal rotation and translation parameters that minimize the geometric error illustrated in Eq. (1) as such:

$$\hat{\mathbf{R}}_i, \hat{\mathbf{t}}_i = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{C}_i} \|\mathbf{q} - \mathbf{R}\mathbf{p} - \mathbf{t}\|, \qquad (4)$$

where $\mathcal{C}_i$ is the i-th triplet. We can retrieve $\hat{\mathbf{R}}_i$, $\hat{\mathbf{t}}_i$ by using Procrustes method [24], a closed-form solution for rigid transformation in SE(3) as follows:

$$\hat{\mathbf{R}}_i = \mathbf{V} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(\mathbf{V}\mathbf{U}^T) \end{bmatrix} \mathbf{U}^T, \qquad (5)$$

where $\tilde{\mathbf{P}}_i\tilde{\mathbf{Q}}_i^T = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ is the singular value decomposition of a covariance matrix of $\tilde{\mathbf{P}}_i, \tilde{\mathbf{Q}}_i \in \mathbb{R}^{3\times3}$, zero-mean shifted coordinates that triplet $\mathcal{C}_i$ forms in each point cloud. The translation parameters are computed as:

$$\hat{\mathbf{t}}_i = \bar{\mathbf{q}}_i - \hat{\mathbf{R}}_i\bar{\mathbf{p}}_i, \qquad (6)$$

where $\bar{\mathbf{p}}_i, \bar{\mathbf{q}}_i \in \mathbb{R}^3$ are the centroids of point cloud $\mathbf{P}_i, \mathbf{Q}_i$. We convert the obtained rotation matrix $\hat{\mathbf{R}}_i$ to its axis-angle representation $\hat{\mathbf{r}}_i$ as shown in Eq. (3).

We then can use the obtained values of $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{t}}_i$ to identify the bin, $\mathbf{v}_i \in \mathbb{N}^6$, in the Hough space for the triplet to vote on, simply by dividing the values of $\hat{\mathbf{r}}_i$ and $\hat{\mathbf{t}}_i$ by their corresponding Hough space bin sizes:

$$\mathbf{v}_i = \left(\left\lfloor \frac{\hat{\mathbf{r}}_i}{b_{\mathbf{r}}} \right\rfloor \odot \left\lfloor \frac{\hat{\mathbf{t}}_i}{b_{\mathbf{t}}} \right\rfloor\right), \qquad (7)$$

where $b_{\mathbf{r}}, b_{\mathbf{t}}$ are the hyperparameters denoting rotation and translation bin sizes, and $\odot$ is the vector concatenation operator. In this work, we simply use 1's as the voting value for each triplet. Note that if we were to utilize the feature similarity values of the correspondences within each triplet for calculating the vote value, our proposed pipeline would be end-to-end trainable in theory.

Defining the bin sizes to be sufficiently small allows accurate predictions to be made, unlike large bin sizes where the range of rotations represented by a single bin is larger. After all the votes have been cast, we end up with a 6D sparse Hough space of transformation parameters, with each bin holding its vote values from the voting process. We refine this Hough space before making the final predictions on the transformation parameters.

3.3. Learning to Refine Hough Space

After all the votes have been cast, the sparse Hough space could be noisy due to triplets containing outlier correspondences. Moreover, the quantization of continuous 6D transformation space into a discrete Hough space may introduce additional noise into the prediction pipeline without any refinement. A simple aid to alleviate the noisy consensus would be to aggregate nearby votes by applying a sim-

ple Gaussian kernel. A Gaussian kernel with an appropriate size may be sufficient to aggregate the local consensus and partially suppress the noisy votes. However, the receptive field would be limited to the kernel size, and it cannot exploit the high-dimensional geometric patterns inherent in the Hough space.

To this end, we devise a high-dimensional sparse convolutional network $f_\theta : \mathbb{R} \to \mathbb{R}$, to refine the Hough space adaptively. Specifically, f_θ is applied on the raw Hough space $\mathcal{H}$ and produces refined Hough space $\mathcal{H}^*$. We use a U-Shaped 6D sparse convolutional network as it can effectively increase the receptive field size for sparse tensors.

$$\mathcal{H}^* = f_\theta(\mathcal{H}). \tag{8}$$

To train f_θ, we supervise it to perform a binary classification task where the true class is the bin which the ground truth transformation matrix falls into.

$$\mathcal{L}_{bce}(\mathcal{H}^*, \mathbf{T}) = \sum_{i=0}^{|\mathcal{H}^*|} \left(y_i \log \mathcal{H}_i^* + (1 - y_i) \log \mathcal{H}_i^{*C} \right), \tag{9}$$

where $\mathbf{T}$ is the ground truth rigid transformation matrix, $y_i = 1$ if i-th bin of $\mathcal{H}^*$ is the bin that $\mathbf{T}$ falls into and otherwise 0, $|\cdot|$ is the cardinality of a sparse tensor, and $\mathcal{H}_i^{*C} = 1 - \mathcal{H}_i^*$. The overview of our proposed pipeline is illustrated in Figure 3.

3.4. Comparison with RANSAC

Here we elaborate on the differences between our method and RANSAC to highlight the design choice of the proposed algorithm. While the motivation of determining the consensus among the candidate correspondences overlap between RANSAC and our method, there are distinct differences. Unlike RANSAC, which adopts *hypothesis-and-verify* pipeline and evaluates the hypothesis for every iteration, our method accumulates the consensus by utilizing the parallel voting procedure and evaluates the results only once after all voting has been finished. This enables fast execution of the pipeline by removing the repetitive verification steps. Furthermore, since our method accumulates the consensus in the form of a 6D sparse tensor, the fully-convolutional refinement network can be applied upon it, leveraging the inherent geometric bias in the accumulated consensus of 6D transformation parameter space. We refer the readers to the Sec.4 for an empirical comparison between RANSAC and our method.

4. Experiments

We conduct experiments with our proposed approach in pairwise point cloud registration scenarios, and we further integrate our method into a multi-way registration pipeline.

4.1. Datasets

For indoor pairwise registration, we use 3DMatch benchmark [56], consisting of point cloud pairs from various real-world indoor scenes. Their ground truth transformations are estimated using RGB-D reconstruction pipelines [15, 25]. While we use the provided train/test split, we follow the standard procedure [13] to generate pairs with an overlap of as low as 30% for both training and testing to evaluate the performance of our method under cases of low overlap ratio. For pairwise registration of outdoor scenes, we use the odometry benchmark of KITTI LIDAR scans [21], consisting of real-world 3D scans taken using Velodyne laser scanners. We follow Choy *et al.* [13] to create pairwise splits for training, validation, and testing. To evaluate our method after integration into a multi-way registration pipeline, we use ICL-NUIM dataset [9] with simulated depth noise following [10].

4.2. Implementation

Throughout our implementation, we extensively use CUDA to perform fast matrix multiplications and parallelized operations for higher efficiency. In particular, the voting process of each correspondence set on all the bins of the Hough space is calculated in parallel. For efficient nearest neighbor search when computing feature matches between point cloud pairs, we use Faiss [32], an opensource library for efficient similarity search and clustering of dense vectors. To utilize sparse tensor representation and perform sparse convolutions, we use MinkowskiEngine [11], an auto-differentiation library for sparse tensors. Open3D [59] is widely used in our experiments for various classical registration methods and visualization of 3D data.

We use the official implementation of FCGF [13] and the provided pre-trained weights for both 3DMatch and KITTI datasets. The Hough space refinement networks were trained for ten epochs with a batch size of 1, using an SGD optimizer with an initial learning rate of 10^{-3} and exponential learning rate decay factor 0.99. We used spatial hashing [46, 59] for 3DMatch and KITTI datasets with the size of 5cm and 30cm, respectively. We use bin sizes of $b_{\mathbf{r}} = 0.02\text{rad}, b_{\mathbf{t}} = 0.02\text{m}$ for 3DMatch dataset, and $b_{\mathbf{r}} = 0.005\text{rad}, b_{\mathbf{t}} = 0.1\text{m}$ for KITTI in our experiments.

While leveraging the feature similarity values of correspondences could enable end-to-end training of our pipeline, we fix our feature extractor network in our experiments. We brought the quantitative results of the baseline traditional and learning-based baseline methods from [10]. For a fair comparison, we re-evaluate the [10] on our environment to alleviate the performance difference due to the different hardware. All the experiments are evaluated on an Intel Xeon Gold 5220R CPU, and an NVIDIA Titan RTX GPU, which is our best approximation of settings to [10].

4.3. Evaluation metric

For all experiments, we report average Relative Rotation Error (RRE) and Relative Translation Error (RTE) as:

$$\text{RRE}(\hat{\mathbf{R}}) = \arccos \frac{\text{Tr}(\hat{\mathbf{R}}^T \mathbf{R}^*) - 1}{2} \qquad (10)$$

$$\text{RTE}(\hat{\mathbf{t}}) = \|\hat{\mathbf{t}} - \mathbf{t}^*\|_2^2, \qquad (11)$$

where $\hat{\mathbf{R}}$, $\hat{\mathbf{t}}$ are the predicted rotation matrix and translation vector, $\mathbf{R}^*$, $\mathbf{t}^*$ are the ground truth, and $\text{Tr}(\cdot)$ indicates the trace of a matrix. Following [10], we report *recall*, which is the ratio of successful pairwise registrations. A pairwise registration is defined as successful if the RRE and RTE are below predefined thresholds. We compute the average RRE and RTE only on successful registrations for better numerical reliability.

However, average RRE and RTE are primarily affected by the recall and do not present an *overall* view of how accurate the predictions are. Therefore, we report the average RTE and RRE regardless of the success of the registration on 3DMatch dataset in Table 2.

4.4. Pairwise Registration

We report pairwise registration results evaluated on the test set of 3DMatch benchmark [56] containing eight different scenes. We compare our methods with both classical methods and learning-based methods, including the current state-of-the-art. In Table 1, we define the thresholds for RRE and RTE of successful registration to be 30cm and 15 degrees, respectively. Our method outperforms all the other techniques in terms of the recall, RRE, and RTE consistently in most scenes, achieving a new state-of-the-art.

Comparison with the traditional methods. As stated in [10], methods of point-to-point and point-to-plane ICP, RANSAC [19], and FGR [58] implemented by Open3D [59] were used. For RANSAC and FGR, FPFH [45] features were used. Furthermore, Go-ICP [55] and Super4PCS [37] were tested using open-source implementations in Python. While the majority of the traditional methods fail in more cases, RANSAC methods demonstrate reasonable results, which improve with an increased number of iterations. Our method is more than three times faster than RANSAC with 2M iterations and achieves significantly higher recall, RRE, and RTE.

Comparison with the learning-based methods. We compare our approach with learning-based pairwise registration methods, namely DCP [52], PointNetLK [2], and DGR [10]. While other methods including 3DRegNet [18] and PRNet [53] exist, they have shown to fail in [10]. DGR offers to be the most effective among the compared methods. They previously achieved the highest recall and the lowest RRE and RTE while almost as fast as FGR. We show

Table 1. Evaluation results on 3DMatch benchmark [56]. The first group of rows shows the results of classical global registration methods, and the second group of rows shows the results of ICP variants. The third group of rows shows the results of learning-based methods. Time includes feature extraction. The reported time of [10] on the original paper was 0.7s, whereas the recalibrated result was 0.96s.

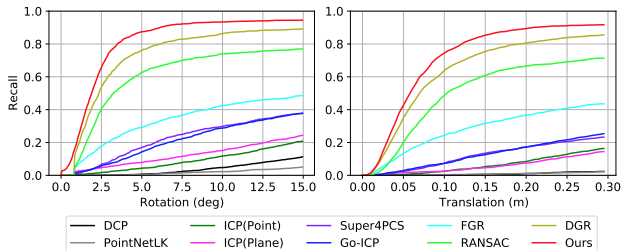|	Recall (%)	RTE (cm)	RRE (°)	Time (s)
FGR [58]	42.7	10.6	4.08	0.31
RANSAC-2M [19]	66.1	8.85	3.00	1.39
RANSAC-4M	70.7	9.16	2.95	2.32
RANSAC-8M	74.9	8.96	2.92	4.55
Go-ICP [55]	22.9	14.7	5.38	771.0
Super4PCS [37]	21.6	14.1	5.25	4.55
ICP(P2Point) [59]	6.04	18.1	8.25	0.25
ICP(P2Plane) [59]	6.59	15.2	6.61	0.27
DCP [52]	3.22	21.4	8.42	0.07
PointNetLK [2]	1.61	21.3	8.04	0.12
DGR [10]	85.2	7.73	2.58	0.70*
Ours	**91.4**	**6.61**	**2.08**	0.46

Figure 4. Overall pairwise registration recall (y-axis) on 3DMatch benchmark with varying rotation (left image) and translation (right image) error thresholds (x-axis) for successful registration. Our approach outperforms baseline methods for all thresholds while being 2× faster than the most accurate baseline.

Table 2. Mean and standard deviation of RTE and RRE on the entire 3DMatch benchmark [56].

	RTE (cm)	RRE (°)
DCP [52]	1.41 ± 0.88	49.48 ± 36.41
PointNetLK [2]	2.28 ± 1.38	86.00 ± 47.70
ICP(Point) [59]	1.07 ± 0.87	35.06 ± 24.91
ICP(Plane) [59]	1.23 ± 4.48	35.32 ± 27.63
Super4PCS [37]	1.59 ± 1.52	64.64 ± 63.79
Go-ICP [55]	1.28 ± 1.27	54.40 ± 58.19
FGR [58]	1.08 ± 1.36	42.27 ± 52.31
RANSAC [19]	0.64 ± 1.18	26.25 ± 49.89
DGR [10]	0.30 ± 0.76	10.80 ± 29.40
Ours	$\mathbf{0.21} \pm 0.67$	$\mathbf{7.95} \pm 27.83$

that our approach outperforms DGR in all evaluation metrics and shows to be around twice as fast as DGR as shown in Table 1 and Table 2. Furthermore, Figure 4 indicates that our method is robust under different rotation and translation thresholds.

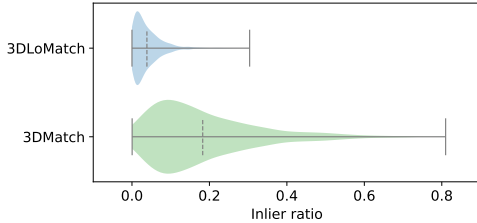Feature extractors such as [4, 13], integrated with

Figure 5. The distribution of inlier ratio of the putative correspondences obtained by feature matching on 3DMatch [56] and 3DLoMatch [30] datasets.

Table 3. Registration recall on 3DLoMatch [30] dataset. The results in the table excluding ours were taken from PointDSC [3]

Feature	Method	Recall(%)
FCGF [13]	RANSAC	35.7
	PointDSC [3]	52.0
	Ours	**56.1**
Predator [30]	RANSAC	54.2
	PointDSC [3]	61.5
	Ours	**64.6**

RANSAC, are also applicable for the pairwise registration task on 3DMatch benchmark. However, they use a different error threshold from ours and are not directly comparable to our method. We include the comparison of our method with the feature extractors + RANSAC under their settings in the supplementary materials.

Robustness to lower inlier ratio. Furthermore, we show that our method is applicable for pairwise registration under even lower overlap ratios (10 to 30%) using 3DLoMatch [30] dataset. Figure 5 shows that the inlier ratio of putative correspondences obtained by feature matching is significantly lower for 3DLoMatch dataset compared to 3DMatch dataset, therefore being a more challenging registration task. The results are shown in Table 3, where our method yields the best results using both FCGF [13] and Predator [30] features. Figure 6 visualizes some of our registration results on 3DLoMatch dataset, where the low overlap ratio is clearly visible.

4.5. Outdoor LiDAR Registration

We evaluate our method on outdoor LIDAR scans from KITTI dataset, following the evaluation protocols of [13]. We use GPS-IMU to create the registration split of the dataset, containing pairs of point cloud fragments that are at least 10m apart. The ground-truth transformations are generated using GPS, which is refined by ICP to fix errors in readings as in [13]. We use the FCGF network trained on the training set of the registration split to extract the pointwise features. We use a voxel size of 30cm for downsampling the point cloud for this experiment since outdoor
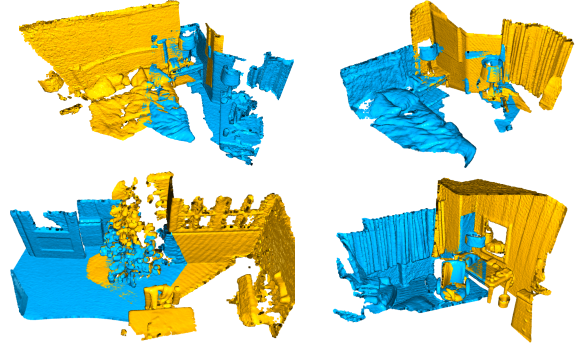

Figure 6. Registration results on 3DLoMatch [30] dataset. As can be also seen in Figure 5, 3DLoMatch is more challenging than 3DMatch due to the lower overlap ratio. Best viewed in color.

Table 4. Evaluation results on KITTI test split [21]. All methods are trained on 30cm voxel downsampled point clouds, and thresholds of 0.6m and $5°$ are used to define successful registration. Time includes feature extraction. The reported running time of [10] on KITTI is 2.29s, where the recalibrated result was 1.86s.

	Recall (%)	RTE (cm)	RRE (°)	Time
FGR [58]	0.2	40.7	1.02	1.42
RANSAC [19]	34.2	25.9	1.39	1.37
FCGF+RANSAC [13]	98.2	**10.2**	0.33	6.38
DGR [10]	96.9	21.7	0.34	2.29*
Ours	**99.1**	19.8	**0.29**	**0.83**

scenes tend to capture an enormous scope of data. The evaluation results are reported in Table 4, where our method shows to outperform the previous state-of-the-art in terms of *recall* and RRE while being more than twice as fast as DGR. FCGF + RANSAC achieves the lowest RTE, but our approach runs seven times faster in comparison. The comparison of our method with the feature extractors [4, 13] + RANSAC under different settings are in the supplementary materials for KITTI dataset as well. The visualization of the registration results is shown in Figure 7.

4.6. Time comparison

While most of the quantitative results on 3DMatch and KITTI datasets are taken from [10], we re-evaluate [10] on our environment for a fair comparison. On 3DMatch dataset, the time reported on [10] is 0.70 seconds on average, while our re-evaluation on our environment yields 0.96 seconds. On outdoor LiDAR registration using KITTI dataset, the time reported on [10] is 2.29 seconds on average, while our re-evaluation on our environment yields 1.86 seconds on average. Therefore, the reported time in [10] is faster than our re-evaluated time on 3DMatch, but slower on KITTI. Our method consistently achieves a faster speed in comparison, demonstrating approximately two times shorter registration times on average.

Figure 7. Registration results of our method on KITTI [21] dataset. Best viewed in color.

Table 5. Ablation tests evaluated on 3DMatch benchmark [56]. Ablations tests are carried out using the Gaussian kernel and the learnable refinement module for two different voxel sizes.

Voxel Size	Smoothing	Refine	Recall (%)	RTE (cm)	RRE (°)
5cm			86.7	7.59	2.45
	✓		89.2	7.00	2.18
		✓	91.4	**6.61**	**2.08**
	✓	✓	**91.5**	6.89	2.23
2.5cm			90.5	6.69	2.03
	✓		91.7	6.25	**1.88**
		✓	**93.7**	**6.23**	1.89
	✓	✓	93.1	6.23	1.94

4.7. Ablation Study

Table 5 illustrates the results of ablation studies on our method regarding 1) the presence of Gaussian smoothing after voting, 2) the presence of the Hough space refinement module, and 3) the choice of voxel size for spatial hashing, It can be seen that both Gaussian smoothing and refining consistently improve the performance for both voxel sizes of 5cm and 2.5cm. However, it can be seen that these two schemes are not complementary, and using both Gaussian smoothing and Hough space refinement together does not constantly improve the results. Applying the refinement module shows higher performance gains than applying Gaussian smoothing; therefore, we choose to use the learnable refinement module alone in our approach.

Using a smaller voxel size of 2.5cm demonstrates consistently superior results than using a larger voxel size of 5cm. We conjecture that this is due to decreased errors derived from the spatial hashing step. Nonetheless, our method sets a new state of the art even when using the larger voxel size of 5cm, proving the efficacy of our approach.

Due to spatial constraints, we refer the readers to the supplementary for additional ablation results regarding our method's execution time and memory requirements.

4.8. Multiway Registration

We show that our model can be integrated into a multiway registration pipeline, exhibiting cross-dataset generalizability. Multi-way registration pipelines of RGB-D scans

Table 6. ATE(cm) on Augmented ICL-NUIM [9].

	FGR [51]	RANSAC [16]	DGR [8]	Ours
Living room1	28.01	23.32	**21.06**	<u>22.91</u>
Living room2	31.71	<u>19.98</u>	21.88	**16.37**
Office1	13.89	<u>13.67</u>	15.76	**12.58**
Office2	15.61	<u>10.95</u>	11.56	**10.90**

leverages pairwise registration to *roughly* align 3D fragments. This is followed by multi-way registration [9] which optimizes the roughly aligned fragment poses with a robust pose graph optimization [33].

We replace the pairwise registration stage in a popular open-source implementation of a multi-way registration pipeline [59] with our proposed method. To illustrate the cross-dataset generalization abilities of our method, we use networks trained on 3DMatch training set while testing on the multi-way registration datasets.

We evaluate the results on the Augmented ICL-NUIM dataset [9] with simulated depth noise for quantitative trajectory results, measuring absolute trajectory error (ATE) as the performance metric. The results can be seen in Table 6, where we outperform state-of-the-art methods on all scenes except Living room 1, where we are second to DGR [10].

5. Conclusion

We present an efficient point cloud registration network for accurate and robust registration of large-scale, real-world 3D scans. Our network pipeline is fast and straight-forward, comprised of dense feature extraction and sparse Hough voting, followed by a learnable refinement module. Features are extracted from a point cloud pair to compute putative correspondences. They are fed to our Hough voting module to vote on a transformation parameter space, constructed in a sparse manner to achieve high accuracy and low memory overhead. We can find the consensus among the correspondences to predict the final transformation parameters. Our sparse Hough voting module is shown to be highly robust against outliers, yielding state-of-the-art results on the indoor scene dataset and comparable results on the outdoor scene benchmark without integrating intensive outlier filtering methods. We believe that integrating keypoint selection to reduce the number of initial correspondences or proposing triplet ranking for a smarter sampling of triplets would be an interesting future research direction for higher efficiency and performance.

References

[1] Dror Aiger, Niloy Mitra, and Daniel Cohen-Or. 4-points congruent sets for robust pairwise surface registration. *35th International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'08)*, 27, 08 2008.

[2] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7163–7172, 2019.

[3] Xuyang Bai, Zixin Luo, Lei Zhou, Hongkai Chen, Lei Li, Zeyu Hu, Hongbo Fu, and Chiew-Lan Tai. Pointdsc: Robust point cloud registration using deep spatial consistency. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15859–15869, 2021.

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

[5] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. International Society for Optics and Photonics, 1992.

[6] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse iterative closest point. *Computer Graphics Forum*, 32, 08 2013.

[7] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017.

[8] Minsu Cho, Suha Kwak, Cordelia Schmid, and Jean Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1201–1210, 2015.

[9] Sungjoon Choi, Qian-Yi Zhou, and V. Koltun. Robust reconstruction of indoor scenes. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565, 2015.

[10] Christopher Choy, Wei Dong, and Vladlen Koltun. Deep global registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2514–2523, 2020.

[11] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[12] Christopher Choy, Junha Lee, Rene Ranftl, Jaesik Park, and Vladlen Koltun. High-dimensional convolutional networks for geometric pattern recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

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

[14] Ondrej Chum and Jiri Matas. Matching with prosac-progressive sample consensus. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 220–226. IEEE, 2005.

[15] Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface re-integration. *ACM Transactions on Graphics 2017 (TOG)*, 2017.

[16] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 602–618, 2018.

[17] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppfnet: Global context aware local features for robust 3d point matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 195–205, 2018.

[18] G Dias Pais, Srikumar Ramalingam, Venu Madhav Govindu, Jacinto C Nascimento, Rama Chellappa, and Pedro Miraldo. 3dregnet: A deep neural network for 3d point registration. *arXiv e-prints*, pages arXiv–1904, 2019.

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

[20] Andrew W Fitzgibbon. Robust registration of 2d and 3d point sets. *Image and vision computing*, 21(13-14):1145–1153, 2003.

[21] Andreas Geiger, Philip Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[22] Zan Gojcic, Caifa Zhou, Jan D Wegner, Leonidas J Guibas, and Tolga Birdal. Learning multiview 3d point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1759–1769, 2020.

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

[24] John C Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

[25] Maciej Halber and T. Funkhouser. Fine-to-coarse global registration of rgb-d scans. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6660–6669, 2017.

[26] A. Handa, T. Whelan, J. McDonald, and A. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, 2014.

[27] Dirk Holz, Alexandru Ichim, Federico Tombari, Radu Rusu, and Sven Behnke. Registration with the point cloud library - a modular framework for aligning in 3-d. *IEEE Robotics and Automation Magazine*, 22:110–124, 12 2015.

[28] Reza Hoseinnezhad and Alireza Bab-Hadiashar. An m-estimator for high breakdown robust estimation in computer vision. *Computer Vision and Image Understanding*, 115(8):1145–1156, 2011.

[29] PVC Hough. Method and means for recognizing complex pattern. 1962.

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

[31] A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *PAMI*, 21(5), 1999.

[32] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2019.

[33] Rainer Kümmerle, Giorgio Grisetti, Hauke Strasdat, Kurt Konolige, and Wolfram Burgard. G2o: A general framework for graph optimization. pages 3607 – 3613, 06 2011.

[34] David Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 11 2004.

[35] Joao Maciel and Joao Costeira. A global solution to sparse correspondence problems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:187– 199, 03 2003.

[36] Haggai Maron, Nadav Dym, Itay Kezurer, Shahar Kovalsky, and Yaron Lipman. Point registration via efficient convex relaxation. *ACM Transactions on Graphics*, 35:1–12, 07 2016.

[37] Nicolas Mellado, Niloy Mitra, and Dror Aiger. Super 4pcs fast global pointcloud registration via smart indexing. *Computer Graphics Forum*, 33, 08 2014.

[38] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2940–2950, June 2021.

[39] Juhong Min, Jongmin Lee, Jean Ponce, and Minsu Cho. Hyperpixel flow: Semantic correspondence with multi-layer neural features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3395–3404, 2019.

[40] Kwang Moo Yi, Eduard Trulls, Yuki Ono, Vincent Lepetit, Mathieu Salzmann, and Pascal Fua. Learning to find good correspondences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2666–2674, 2018.

[41] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3d object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[42] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[43] Rahul Raguram, Ondrej Chum, Marc Pollefeys, Jiri Matas, and Jan-Michael Frahm. Usac: a universal framework for random sample consensus. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):2022–2038, 2012.

[44] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[45] R. B. Rusu, N. Blodow, and M. Beetz. Fast point feature histograms (FPFH) for 3D registration. In *ICRA*, 2009.

[46] R. B. Rusu, Nico Blodow, and M. Beetz. Fast point feature histograms (fpfh) for 3d registration. *2009 IEEE International Conference on Robotics and Automation*, pages 3212–3217, 2009.

[47] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz. Aligning point cloud views using persistent feature histograms. In *IROS*, 2008.

[48] S. Salti, F. Tombari, and L. di Stefano. SHOT: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125, 2014.

[49] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. *Comput. Graph. Forum*, 26:214–226, 06 2007.

[50] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In *European conference on computer vision*, pages 356–369. Springer, 2010.

[51] F. Tombari, S. Salti, and L. Di Stefano. Unique shape context for 3d data description. In *ACM Workshop on 3D Object Retrieval*, 2010.

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

[53] Yue Wang and Justin M Solomon. Prnet: Self-supervised learning for partial-to-partial registration. In *Advances in Neural Information Processing Systems*, pages 8814–8826, 2019.

[54] Heng Yang and Luca Carlone. A polynomial-time solution for robust registration with extreme outlier rates. *arXiv preprint arXiv:1903.08588*, 2019.

[55] Jiaolong Yang, Hongdong li, Dylan Campbell, and Yunde Jia. Go-icp: A globally optimal solution to 3d icp point-set registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38:1–1, 12 2015.

[56] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *CVPR*, 2017.

[57] Jiahui Zhang, Dawei Sun, Zixin Luo, Anbang Yao, Lei Zhou, Tianwei Shen, Yurong Chen, Long Quan, and Hongen Liao. Learning two-view correspondences and geometry using order-aware network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5845–5854, 2019.

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

[59] Qian-Yi Zhou, Jaesik Park, and V. Koltun. Open3d: A modern library for 3d data processing. *ArXiv*, abs/1801.09847, 2018.