

# Category-Level Global Camera Pose Estimation with Multi-Hypothesis Point Cloud Correspondences

Jun-Jee Chao<sup>1</sup>, Selim Engin<sup>1</sup>, Nicolai Häni<sup>1</sup> and Volkan Isler<sup>1</sup>

**Abstract**—Correspondence search is an essential step in rigid point cloud registration algorithms. Most methods maintain a single correspondence at each step and gradually remove wrong correspondences. However, building one-to-one correspondence with hard assignments is extremely difficult, especially when matching two point clouds with many locally similar features. This paper proposes an optimization method that retains all possible correspondences for each keypoint when matching a partial point cloud to a complete point cloud. These uncertain correspondences are then gradually updated with the estimated rigid transformation by considering the matching cost. Moreover, we propose a new point feature descriptor that measures the similarity between local point cloud regions. Extensive experiments show that our method outperforms the state-of-the-art (SoTA) methods even when matching different objects within the same category. Notably, our method outperforms the SoTA methods when registering real-world noisy depth images to a template shape by up to 20% performance.

## I. INTRODUCTION

Given an image of an object from a known category, computing the camera pose with respect to the object frame is a challenging problem in robot localization [1], robotic manipulation [2] and object inspection [3]. This problem is even more challenging when we are given neither two views of the same scene nor the exact 3D model corresponding to the object. In our setting, the template shape can be significantly different from the observed object in terms of its geometry *and* appearance. For example, it can be a simple synthetic model of a car that is significantly different from the observed real car.

To solve this novel pose estimation problem, new methods are needed: Traditional Structure from Motion (SfM) explores visual features like SIFT [4] to establish correspondences between sparse views, but some overlap is required. Moreover, these features rely heavily on visual similarity. However, in our case, the observed object might have a completely different texture than the template shape from the synthetic domain. Point cloud registration methods such as the Iterative Closest Point (ICP) algorithm [5], [6] provide an alternative way to estimate the camera pose by avoiding the domain gap introduced by texture. However, ICP is sensitive to initialization and can get trapped in local minima. Point feature descriptors are a way to mitigate the initialization and local minima problem by matching keypoints with distinctive local features [7]–[9].

Recently, learning-based methods have been proposed to address the limitations of these traditional methods. Instead of requiring view overlap for feature matching, deep

learning approaches learn a direct mapping from images to poses [10]–[12]. More recently, deep learning approaches have been used to build 2D-3D correspondences [13]–[16]. Nevertheless, these learning-based methods suffer from out-of-distribution samples during test time and are sensitive to texture differences. Besides learning from textures, deep learning is used to improve traditional point feature descriptors [17]–[24]. However, learning-based methods require a large amount of labeled data which is not always applicable for real-world applications. Moreover, these methods usually require training on a specific category or dataset. In contrast, our proposed method requires only one point cloud from the target category, and is less sensitive to geometry difference and noise. We show that our method generalizes easily across different datasets and demonstrate its direct applicability to robotic grasping as a representative task.

Our work follows the classical pipeline of building point correspondences with local features but adds a number of novel ideas. A fundamental limitation of existing optimization-based methods is that they build one-to-one correspondences between the observed partial point cloud and the complete point cloud. However, the point distribution between the complete and partial point clouds differs, making matching between local descriptors nontrivial. Moreover, our observed point cloud and the given template shape can have different geometry, which makes building one-to-one correspondences in a single shot more difficult. Our method addresses these issues as follows:

- Instead of building one-to-one matching between keypoint features, we consider multiple hypotheses and design an optimization method to handle one-to-many soft feature assignments.
- To account for different point distributions between the complete and partial point clouds, we introduce a partitioning scheme to register the observed partial depth scans to the provided template shape together with a novel keypoint descriptor that captures point distribution statistics in a local neighborhood.
- We show that our proposed approach can act as a drop-in replacement to existing feature matching methods and improves performance by up to 50% on synthetic data.
- We show that our method achieves comparable results with learning-based methods on synthetic data and outperforms them by up to 20% on noisy real-world data. Moreover, without requiring any training, our method generalizes better across different datasets and objects.

Experiments on synthetic data, real-world benchmarks, and robotic applications are conducted to justify our claims.

<sup>1</sup> Department of Computer Science, University of Minnesota, USA  
{chao0107, engin003, haeni001, isler}@umn.edu

## II. RELATED WORK

This section reviews existing approaches for point cloud registration and category-level object pose estimation.

**Point cloud registration:** The classical ICP [5], [6] iteratively solves for the relative pose between two point clouds by alternating between closest point search and transformation estimation. Due to its myopic nature, it is well known that ICP can get stuck in local minima, especially when the two point-sets are not identical. Although newer variants have been proposed [25]–[27], these methods are still sensitive to pose initialization. Hand-crafted point cloud feature descriptors, like PFH [7], FPFH [8], promise to overcome the initialization issue by establishing correspondences between descriptive features. FGR [9] achieves global registration by optimizing a correspondence-based objective function using point feature descriptors. TEASER [28], [29] applies a graph-theoretic framework to decouple the pose estimation problem into sub-problems which allows them to solve for point clouds with a large fraction of noisy correspondences. However, these optimization-based methods focus on building one-to-one correspondences, which can be ambiguous for similar feature descriptors. In contrast, our one-to-many matching is less affected by such ambiguities.

Recently, deep learning methods learn point feature descriptors for correspondence establishment [18], [19], [30]–[32]. Many methods apply PointNet [33], [34] architecture to extract features from raw point clouds [24]. For example, PPFNet [30] learns feature descriptors from the point distribution in a local neighborhood. DCP [19] and PRNet [35] apply DGCNN for feature learning. In addition to learning feature extraction, RPM-Net [20] learns to predict optimal annealing parameters to obtain soft correspondences. More work have been developed specifically for local partial-to-partial point cloud registration [21]–[23], [36], [37]. Some deep learning methods supervise training solely using ground truth point correspondences, without access to the ground truth transformation matrix [23], [38]–[40]. In contrast, our proposed method does not have any trainable parameters, thus, does not require a large amount of annotated data.

**Category-level object pose estimation** Next to work that tries to match point clouds of the same underlying object, the second line of work proposes to align shapes in a given category to a common, learned coordinate frame. Category-level object pose estimation methods achieve impressive results when trained with supervision [41]–[43], and using only self-supervision [44]–[48]. Canonical Capsule [45] uses an auto-encoder framework to learn pose-invariant capsules to represent object parts, which can be further used to reconstruct the object in the learned canonical frame. However, Canonical Capsules only work with complete point clouds. Equi-pose [46] leverages a SE(3) equivariant network to estimate the object pose and learns an invariant shape reconstruction module that works on both partial and complete point clouds. While these methods achieve impressive results on synthetic datasets, our experiments show that they do not work well on noisy real-world scans.

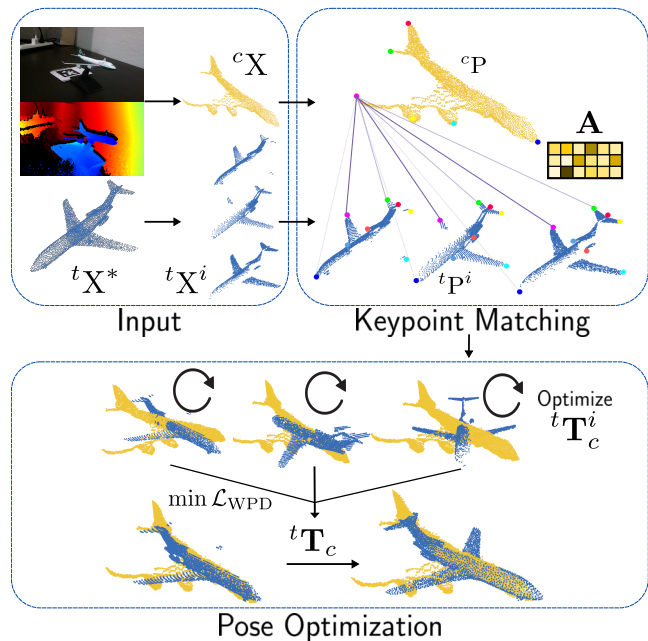


Fig. 1: **Method Overview.** Given a partial point cloud  ${}^cX$  and a complete template point cloud  ${}^tX^*$ , we first partition the template into  $m = 18$  partial point clouds  ${}^tX^i$ . Our method samples keypoints using farthest point sampling and generates an affinity matrix  $A$  that matches each keypoint to points in the partial templates. Using this affinity matrix, we jointly optimize the matching loss and the correspondence matrix to obtain better matches. The optimal pose is the one that minimizes the matching loss between the observed and the partial template point clouds.

## III. METHOD

We formulate the camera pose estimation problem as matching an observed partial point cloud  ${}^cX$  in the camera frame  $\{c\}$  to a complete template point cloud  ${}^tX^*$  from the same category as the observed object, in the template’s object frame  $\{t\}$ . In our notation, the  $*$  superscript denotes the complete point cloud. We assume known point normals (which can be computed from the depth images [49] or through local plane fitting [50]), and known ground normal (which can be estimated by upright orientation estimation [51]). In practice, however, we found it sufficient to estimate the ground normal by fitting a plane to the masked background point cloud. Given the observed and complete template point clouds, our objective is to estimate the rigid transformation  ${}^tT_c \in SE(3)$  that transforms the observed point cloud in the camera frame to the template frame.

### A. Overview of our approach

Given an object’s depth image, we generate a partial point cloud  ${}^cX$  by lifting the depth pixels into 3D using the known camera intrinsics. Given this partial point cloud in the camera frame and a template point cloud  ${}^tX^*$ , we estimate the rigid transformation  ${}^tT_c$  by following the classical two-step pipeline: correspondence search and pose estimation. Our approach is visualized in Fig. 1. We first create a dictionary of  $m$  partial template point clouds  ${}^tX^i \in \mathcal{X}$ , where  $\mathcal{X} = \{{}^tX^i | i = 1, \dots, m\}$  is constructed by removing invisible points of  ${}^tX^*$  from  $m$  viewpoints. Partitioning is necessary, as feature descriptors on the partial point cloud

will not match the descriptors on the complete template due to differences in neighborhood point distributions. Note that this dictionary generating process differs from pose initialization as the partial template point clouds have the same orientation in the template frame; only different parts are missing. Next, we extract keypoints  ${}^c\mathbf{P}$  and  ${}^t\mathbf{P}^i$  from the partial input and the partial template point clouds, respectively. We extract feature descriptors from the point clouds and construct affinity matrices to match the input point cloud  ${}^c\mathbf{X}$  to a set of partial point clouds  ${}^t\mathbf{X}^i$ . This matching process generates  $m$  pose candidates. We jointly optimize the pose estimate and the correspondence matrix by minimizing the weighted Euclidean distance between keypoints. We select the final pose estimation as the one that minimizes the matching loss between the partial input point cloud and the template. Next, we describe each component in detail.

### B. Affinity matrix from feature matching

To match a partial point cloud to each partitioned template in the dictionary, we aim to establish an affinity matrix  $\mathbf{A}$  which indicates the pairwise similarity between the point feature descriptors. First, we extract keypoints  ${}^c\mathbf{P} \in \mathbb{R}^{3 \times n}$ ,  ${}^t\mathbf{P}^i \in \mathbb{R}^{3 \times 2n}$  from the input and the partial template point cloud respectively using farthest point sampling (FPS) [34].

We also require expressive feature descriptors to match the input to the partial templates. In this paper, we provide two options for the point feature descriptor. The first is the Point Feature Histogram (PFH) [7], a commonly used descriptor in the literature, and the second one is a new descriptor that we introduce, named Local Patch Similarity (LPS).

**Point Feature Histogram (PFH):** PFH is a histogram-based, rotation-invariant point feature descriptor that measures the curvature property within the local neighborhood of the keypoint based on the point normals. Once we have extracted PFH descriptors, we can match them by applying Earth Mover’s Distance (EMD) [52] to measure their similarity. Therefore, the affinity score given by the PFH descriptor between two keypoints is:

$$\mathbf{A}_{jk} = 1 / (\text{EMD}(f_{PFH}({}^c\mathbf{P}_j), f_{PFH}({}^t\mathbf{P}_k)) + \epsilon) \quad (1)$$

where  $\epsilon$  is a small value to avoid zero division.

**Local Patch Similarity (LPS):** For our second descriptor, we introduce the novel LPS descriptor. Given a point cloud  $\mathbf{X}$ , keypoint  $p \in \mathbf{X}$ , point normal  $\vec{n}_p$ , ground normal  $\vec{n}$ , and a feature radius  $r$ , we first define a local reference frame as shown in Fig. 2. The  $x$ -axis of the local reference frame is defined as the point normal  $\vec{x} = \vec{n}_p$ ,  $y$ -axis is defined as  $\vec{y} = \vec{n} \times \vec{x}$ , and  $z$ -axis is defined as  $\vec{z} = \vec{x} \times \vec{y}$ . In the case when  $\vec{n}$  is aligned with  $\vec{n}_p$ , we apply PCA using the points within the neighborhood of  $p$  and set  $\vec{x}$  to be the eigenvector with the smallest eigenvalue. Finally, we crop all the points with a distance to  $p$  less than the feature radius  $r$  and transform them into the local reference frame.

To measure the similarity between two keypoints, we calculate the F-score [53] between two local patches. Therefore, the affinity score given by our feature descriptor is:

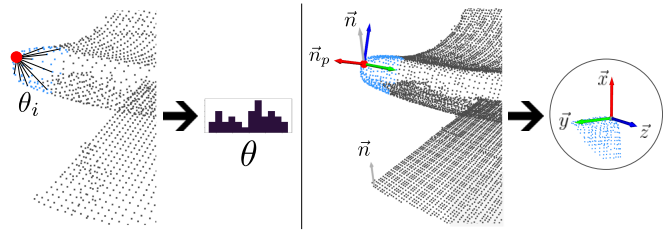


Fig. 2: The **Point Feature Histogram (PFH)** and **Local Patch Similarity (LPS)** descriptors. PFH (left) uses a histogram of angles to capture the pairwise geometrical properties between all points within a keypoint’s neighborhood. LPS (right) constructs a local coordinate frame for the neighborhood of a keypoint, and computes the similarity of two keypoints using the F-score between them.

$$\mathbf{A}_{jk} = \text{F-score}(f_{LPS}({}^c\mathbf{P}_j), f_{LPS}({}^t\mathbf{P}_k)) \quad (2)$$

Given these two feature descriptors, we can create our affinity matrix  $\mathbf{A} \in \mathbb{R}^{n \times 2n}$  by either using one or the other descriptors’ affinity scores. Finally, we normalize the affinity matrix  $\mathbf{A}$  such that each row sums to 1.

### C. Initial pose estimation using SVD

The next step is to build pairing points between the two frames for pose initialization. We assign a corresponding point in the template frame for each keypoint in the partial point cloud by weighting all the template keypoints with the associated affinity score. Since the affinity matrix is normalized such that each row sums to 1, the soft assignment can be achieved by  $\mathbf{A} \cdot {}^t\mathbf{P}^\top$ . In other words, we *softly* assign a corresponding virtual point in the template frame for each point in  ${}^c\mathbf{P}$  with the noisy affinity matrix  $\mathbf{A}$ . Following the method introduced by [19], [54], the initial pose can be solved as:

$$\mathbf{U}\Sigma\mathbf{V}^\top = \text{SVD}({}^c\bar{\mathbf{Q}} \cdot {}^t\bar{\mathbf{Q}}^\top) \quad (3)$$

$${}^t\mathbf{R}_c = \mathbf{V}\mathbf{U}^\top \text{ and } {}^t\vec{t}_c = -{}^t\mathbf{R}_c {}^c\bar{\mathbf{Q}} + {}^t\bar{\mathbf{Q}} \quad (4)$$

where  ${}^c\bar{\mathbf{Q}}$  is the translation normalized  ${}^c\mathbf{P}$  and  ${}^t\bar{\mathbf{Q}}^\top$  is the translation normalized  $\mathbf{A} \cdot {}^t\mathbf{P}^\top$ .  ${}^t\mathbf{R}_c$  and  ${}^t\vec{t}_c$  are the rotation and translation components of the estimated pose  ${}^t\mathbf{T}_c$ . The sign of the diagonal matrix  $\Sigma$  accounts for the choice of reversing the orientation.

### D. Optimize the pose

We first update the estimated pose  ${}^t\mathbf{T}_c$  to minimize the following loss function:

$$\mathcal{L}_{\text{WPD}}(\mathbf{A}, {}^t\mathbf{T}_c) = \sum \mathbf{A} \odot d({}^t\mathbf{T}_c \cdot {}^c\mathbf{P}, {}^t\mathbf{P}) \quad (5)$$

where  $d(\cdot, \cdot)$  denotes the pairwise Euclidean distance between two point sets, and  $\odot$  denotes the element-wise matrix multiplication. This loss function measures the weighted pairwise distance between  ${}^c\mathbf{P}$  and  ${}^t\mathbf{P}$  such that after transforming keypoints  ${}^c\mathbf{P}$  into the template frame, the pair of keypoints  ${}^c\mathbf{P}_j$  and  ${}^t\mathbf{P}_k$  with higher affinity score  $\mathbf{A}_{jk}$  matters more. Furthermore, we penalize the estimated pose that moves point pairs with high-affinity scores far away from each other.

The second step of this optimizer is to simultaneously update the estimated pose  ${}^t\mathbf{T}_c$  and the correspondence matrix  $\mathbf{C} \in \mathbb{R}^{n \times 2n}$  where each element  $\mathbf{C}_{jk}$  indicates the level of correspondence between  ${}^c\mathbf{P}_j$  and  ${}^t\mathbf{P}_k$ . We initialize  $\mathbf{C}$  as the affinity matrix from the previous step. The loss function now becomes:

$$\mathcal{L}_{\text{WPD}}(\mathbf{C}, {}^t\mathbf{T}_c) = \sum \mathbf{C} \odot d({}^t\mathbf{T}_c \cdot {}^c\mathbf{P}, {}^t\mathbf{P}) \quad (6)$$

With this optimization step, we can gradually strengthen the correspondence between the correct matching pair of  ${}^c\mathbf{P}_j$  and  ${}^t\mathbf{P}_k$  by considering the cost of all point pairs under the same rigid transformation.

### E. Dictionary matching

After matching the partial point cloud  ${}^c\mathbf{X}$  to  $m$  point clouds in the template dictionary  $\mathcal{X}$ , the final pose is selected as the pose that has the minimum loss after the final optimization step.

## IV. EXPERIMENTS

Before showing our result on real-world noisy scans, and our direct applicability to robotic manipulation in Section IV-C and IV-D, we design experiments that investigate the limitations of the existing approaches. In Section IV-B we start by registering synthetic partial point clouds to the ground truth complete point cloud, and show in Table I how our dictionary matching pipeline improves the baselines compared to directly matching the partial to the complete point cloud. Next, we consider the case when no ground truth complete shapes are available. Therefore, we match the whole synthetic dataset to category-level representative shapes and show that our method outperforms the optimization-based baselines and achieves comparable results with the learning-based methods. In Section IV-C and Fig. 3, we present results on challenging noisy real-world scans of both outdoor objects and tabletop objects where no ground truth models are available. We demonstrate that our method outperforms the baselines on a variety of objects and show that optimization-based methods are sometimes more practical in real-world applications when no training process is needed. Finally, we apply our method to real-world robotic grasping and demonstrate that we are able to pick up objects like a mug, cap, can, and headset.

### A. Setup

**Datasets.** We conduct experiments on a variety of datasets, including a synthetic dataset ModelNet40 [55], a real-world scan benchmark of tabletop objects TYO-L [56], and tabletop objects scanned by our manipulator. For ModelNet40 [55] and TYO-L [56], we evaluate all the methods on the official testing split. We also collect a real-world dataset containing three toy airplane models, three outdoor cars, and a chair captured with Intel® RealSense™ cameras. To include more samples, 10 RGBD videos of different chairs from Redwood-3dscan [57] are added to this dataset resulting in 17 objects.

**Implementation details.** We use the point normal estimation from Open3D [58] and the farthest point sampling (FPS) implementation from PyTorch3D [59] for keypoint extraction

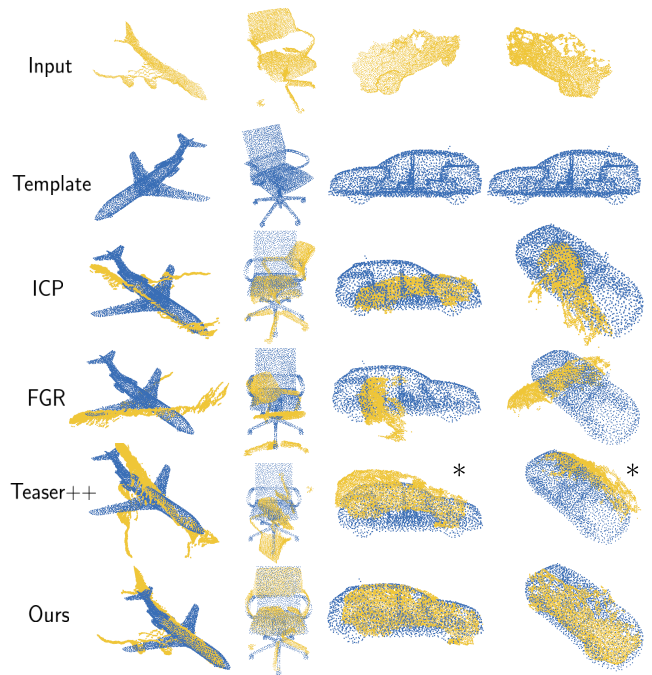


Fig. 3: Qualitative results of partial point cloud registration to templates using noisy real-world depth scans. The associated quantitative results can be found in Table IV. Results marked with \* use the affinity matrix computed by our LPS descriptor as TEASER++ did not converge for this car scan with PPFH.

and point cloud down-sampling. We use PyTorch’s [60] Adam optimizer [61] to minimize the cost function by updating the estimated pose and correspondence matrix. We set the learning rate to 0.001 and run 300 steps for each optimization step. We use the continuous 6D representation [62] to represent rotations.

**Baselines.** We compare against optimization-based methods: ICP [5], FGR [9], TEASER++ [28], learning-based partial-to-partial registration method: RPM-Net [20], and learning-based canonicalization method: Equi-pose [46]. For ICP and learning-based methods, we do not report the translation as we shift the input partial point cloud to the zero-mean center. To provide a good initialization for ICP, we divide the full  $\text{SO}(3)$  into 20 sub-regions and sample one rotation from each sub-region. The learning-based methods are trained on the official training split of ModelNet40 and evaluated on both synthetic and real data of the seen categories.

**Metrics.** We perform pose evaluation and report the rotation error in the form of mean, median,  $10^\circ$  and  $30^\circ$  accuracy. Given the ground truth pose  $\{\mathbf{R}_{gt}, \vec{t}_{gt}\}$  and the estimated pose  $\{\mathbf{R}_{pred}, \vec{t}_{pred}\}$ , the rotation error is defined as:

$$\mathcal{L}_R(\mathbf{R}_{gt}, \mathbf{R}_{pred}) = \arccos \frac{(\text{Tr}(\mathbf{R}_{gt}^T \mathbf{R}_{pred}) - 1)}{2} \quad (7)$$

where  $\text{Tr}$  is the matrix trace. The translation error is defined as:

$$\mathcal{L}_T(\vec{t}_{gt}, \vec{t}_{pred}) = \|\vec{t}_{gt} - \vec{t}_{pred}\|_2 \quad (8)$$

For all the tables in the following sections, we denote  $\mathcal{L}_T$  as the mean translation error for notational simplicity. The best

TABLE I: Partial point cloud registration with ground truth complete shape on ModelNet40. Comparison of directly matching the partial to complete point cloud and our proposed dictionary matching with different feature descriptors.

Pipeline / Descriptor	Methods	Airplane		Chair		Car	
		Mean <sup>o</sup> ↓	Med. <sup>o</sup> ↓	Mean <sup>o</sup> ↓	Med. <sup>o</sup> ↓	Mean <sup>o</sup> ↓	Med. <sup>o</sup> ↓
-	Equi-pose	3.000	1.753	13.717	3.058	97.501	174.271
	ICP	44.992	<b>0.189</b>	113.896	172.167	89.762	81.487
Direct match	FGR	52.792	24.892	91.763	92.960	77.701	79.109
	FPFH	46.251	20.657	90.609	94.106	65.424	42.650
Dictionary	FGR	4.054	1.235	13.016	2.814	71.380	26.895
	FPFH	3.354	1.066	9.496	2.084	63.674	22.393
Dictionary LPS	FGR	22.538	1.181	51.911	12.665	61.975	41.239
	TEASER++	3.699	0.485	11.384	2.113	<b>30.656</b>	<b>4.886</b>
	Ours	<b>1.108</b>	0.341	<b>4.759</b>	<b>0.997</b>	39.936	6.796

result is in **bold** and the second best is underscored.

### B. Point cloud registration on synthetic data

First, we register a partial point cloud to the ground truth model of the same object. As shown in the first four rows of Table I, our method outperforms all the other original optimization-based baselines by a large margin. Equi-pose performs well on synthetic data, especially in the airplane and chair categories. Similar to the observation from the original paper [46], it predicts many 180-degree flips when categories are more symmetric, such as in the car category. The tuned ICP achieves a lower median error than ours on the airplane category while having a larger mean error. As expected, ICP achieves a good matching result when the initialization is close to the target pose. Otherwise, it converges to local minima, which leads to large errors. We apply the LPS feature descriptor to our method for this task, while the FGR and TEASER++ use the FPFH descriptor for building correspondences between the partial input and the complete point cloud.

**Difficulties of directly matching partial to complete point clouds.** In Section III-A we argue that the difference in point distribution between the partial and complete point cloud affects the descriptors, motivating our proposed template partitioning pipeline. We justify the need for our pipeline by applying it to FGR and TEASER++ with the original FPFH descriptor. We estimate the transformations between the partial input and all partitioned templates instead of directly registering to the complete shape. Then we select the estimated pose that minimizes the one-directional Chamfer distance from the input to the complete template. As shown in row 5 and 6 of Table I, adopting our pipeline significantly improves the performance of FGR and TEASER++.

We further replace the FPFH feature descriptor with our LPS descriptor such that the same affinity matrix is initialized for all three methods. Since the two baselines take only one-to-one correspondences, we assign the correspondences with point pairs of the highest score in the affinity matrix. As shown in Table I, our LPS descriptor achieves comparable results with the FPFH descriptor, even with sparse correspondences. Although the point pairs with the highest affinity score given by our LPS descriptor are more likely to be the correct matches, these baselines do not consider multiple hypotheses, which leads to wrong pose estimates if the initial descriptor matching is inaccurate.

**Category-level template matching.** Next, we consider a more realistic scenario where no instance-level ground truth

TABLE II: Partial point cloud registration with one category-level template on ModelNet40 airplane.

Pipeline / Descriptor	Methods	Airplane				
		Mean <sup>o</sup> ↓	Med. <sup>o</sup> ↓	@10 ↑	@30 ↑	$\mathcal{L}_T$ ↓
-	Equi-pose	<b>3.000</b>	<b>1.753</b>	<b>0.977</b>	<b>0.996</b>	-
	ICP	95.93	92.3468	0.327	0.413	-
Direct match	FGR	106.056	102.37	0.002	0.035	1.051
	FPFH	46.296	19.499	0.221	0.665	0.552
Dictionary	FGR	29.831	9.034	0.535	0.819	0.355
	FPFH	27.766	8.112	0.594	0.863	<u>0.337</u>
Dictionary PFH	Ours	<u>20.954</u>	<u>4.735</u>	<u>0.824</u>	<u>0.902</u>	<b>0.263</b>

point cloud is available. Therefore, we register the partial input to a template shape chosen from the same category. In Table II, we report the results of matching the entire airplane category to only one template shape. Compared to matching with the ground truth point clouds, we can see a performance drop for all the optimization-based methods. However, we achieve comparable results with the learning-based method Equi-pose. Moreover, we again demonstrate how our matching pipeline improves the baselines in this challenging task. Therefore, we adopt our matching pipeline to FGR and TEASER++ for the following experiments.

We further investigate how multiple templates can help improve the performance of matching. We focus on the chair category, which has the largest shape variety. To select templates, we apply K-means clustering on the training set using the F-score metric and select five representative chair models as our templates. Then the entire test set is registered to these five templates. We report the minimum matching error across templates for all methods. As shown in Table III, we can see how all optimization-based methods benefit from adequately selected templates. Moreover, our method outperforms the other optimization-based baselines and achieves comparable results with the learning-based method in this challenging setup.

### C. Camera pose estimation on real-world noisy scans

For real-world applications where the object frame is usually unknown, we are interested in estimating the relative camera pose between two depth scans by matching each view to the synthetic category-level template shape. We present the result on our noisy scans of airplane models, chairs, and cars in Table IV and Fig. 3. Another learning-based baseline: RPM-Net, which directly learns partial-to-partial point cloud registration, is also included in this experiment. Since the selected car template is similar to the observed cars, we

TABLE III: Partial point cloud registration with different numbers of category-level templates on ModelNet40 chair. † PFH descriptor with dictionary. ‡ FPFH descriptor with dictionary.

Number of templates	Methods	Chair			
		Mean <sup>o</sup> ↓	Med. <sup>o</sup> ↓	@10 ↑	@30 ↑
-	Equi-pose	<b>13.717</b>	<b>3.058</b>	<b>0.837</b>	<b>0.957</b>
	ICP	129.095	169.172	0.071	0.161
	FGR <sup>‡</sup>	100.696	104.995	0.049	0.14
	TEASER++ <sup>‡</sup>	96.019	103.134	0.077	0.239
1	Ours <sup>†</sup>	41.289	15.556	0.33	0.688
	ICP	51.872	32.804	0.351	0.483
	FGR <sup>‡</sup>	54.930	35.536	0.1	0.439
	TEASER++ <sup>‡</sup>	25.793	16.401	0.290	0.725
5	Ours <sup>†</sup>	<u>19.520</u>	<u>10.664</u>	<u>0.467</u>	<u>0.854</u>

TABLE IV: Pairwise relative pose estimation between two input partial point clouds on real-world scenarios. Our method applies the PFH descriptor on the airplane and LPS descriptor for chair and car based on experimental results.

Methods	ICP	FGR	TEASER++	Equi-pose	RPM-Net	Ours
Airplane	Mean <sup>o</sup> ↓	70.194	104.593	75.116	47.446	<b>37.032</b>
	Med. <sup>o</sup> ↓	<b>7.404</b>	114.430	64.788	<b>5.740</b>	16.155
	@10 ↑	<b>0.552</b>	0.018	0.026	<b>0.710</b>	0.261
	@30 ↑	0.617	0.110	0.265	<b>0.744</b>	<b>0.776</b>
Chair	Mean <sup>o</sup> ↓	108.509	109.612	111.902	112.581	<b>28.026</b>
	Med. <sup>o</sup> ↓	118.724	117.199	121.589	120.814	<b>13.424</b>
	@10 ↑	<b>0.153</b>	0.005	0.019	0.030	<b>0.333</b>
	@30 ↑	<b>0.203</b>	0.080	0.111	0.077	<b>0.836</b>
Car	Mean <sup>o</sup> ↓	111.746	79.185	<b>65.832</b>	96.880	<b>6.643</b>
	Med. <sup>o</sup> ↓	149.675	35.478	<b>16.470</b>	173.236	<b>126.713</b>
	@10 ↑	0.087	0.231	0.256	<b>0.391</b>	<b>0.106</b>
	@30 ↑	0.240	0.482	<b>0.607</b>	0.454	<b>0.228</b>

observe that adding an additional one-directional Chamfer distance to (6) aids the performance. Despite the noise in the depth images, the tuned ICP achieves a lower median rotation error in the airplane category but has a higher mean error. Again, this shows that ICP relies heavily on initialization and is prone to local minima. Equi-pose achieves the lowest median in the airplane category but performs worse in other categories. We also observe that RPM-Net performs better on image pairs with small relative transformations but does not work well on the full dataset. In contrast, our method is more stable for all objects when given a similar template shape to the scan.

**Tabletop objects.** Five categories: bottle, bowl, can, plate, and mug from the TYO-L [56] dataset are included in this experiment. For each category, we select one object from the Shapenet [63] dataset as the template shape. We report the error in the alignment of the main axis for symmetric objects and relative pose error for asymmetric objects. As shown in Table V, our method outperforms the baselines on most objects, while all the feature descriptor-based methods perform similarly on the bottle. We observe that the handles of the mugs are sometimes invisible or too noisy such that all the methods perform poorly on the mug category. However, in the next section, we show that we are able to match the body part of the mug when given a cleaner scan. Finally, we present the datasets statistics for all objects in Table VI.

#### D. Grasping application

We conduct experiments using a Kinova Gen3 manipulator with a RealSense RGBD camera mounted on the arm. Four tabletop objects: a cap, can, mug, and headset, are included in this experiment. For each object, we select a synthetic

TABLE V: Camera pose estimation on the TYO-L dataset. Our method applies PFH descriptor as objects are not fixed on the table in this dataset.

Methods	Objects	Bottle	Bowl	Can	Plate	Mug
ICP	Mean <sup>o</sup> ↓	106.087	88.892	22.960	25.915	124.480
	Med. <sup>o</sup> ↓	154.767	121.756	17.249	<b>4.162</b>	131.630
	@10 ↑	0.0	0.125	0.469	0.781	<b>0.005</b>
	@30 ↑	0.375	0.344	0.563	0.875	0.039
FGR	Mean <sup>o</sup> ↓	<b>28.541</b>	57.287	43.718	80.613	123.431
	Med. <sup>o</sup> ↓	5.896	53.424	53.608	61.498	130.472
	@10 ↑	<b>0.75</b>	0.063	0.188	0.063	<b>0.008</b>
	@30 ↑	<b>0.813</b>	0.25	0.344	0.188	0.031
TEASER++	Mean <sup>o</sup> ↓	35.181	50.812	14.127	73.672	<b>116.964</b>
	Med. <sup>o</sup> ↓	<b>3.202</b>	43.725	10.485	22.000	<b>124.381</b>
	@10 ↑	<b>0.813</b>	<b>0.156</b>	0.438	0.156	0.001
	@30 ↑	<b>0.813</b>	0.375	0.906	0.563	<b>0.046</b>
Ours	Mean <sup>o</sup> ↓	36.556	<b>36.542</b>	<b>9.506</b>	<b>15.630</b>	121.215
	Med. <sup>o</sup> ↓	<b>5.871</b>	<b>31.999</b>	<b>8.398</b>	4.797	<b>129.402</b>
	@10 ↑	0.75	<b>0.156</b>	<b>0.625</b>	<b>0.906</b>	0.002
	@30 ↑	<b>0.813</b>	<b>0.438</b>	<b>1.0</b>	<b>0.938</b>	0.025

TABLE VI: Real-world datasets statistics. We measure the pair-wise relative pose for asymmetric objects and the relative angle between the main axis in the camera frame and the ground truth object frame for symmetric objects.

	Airplane	Chair	Car	Bottle	Bowl	Can	Plate	Mug
Mean <sup>o</sup>	94.446	73.954	93.315	78.117	88.146	59.598	97.346	125.830
Med. <sup>o</sup>	94.240	94.073	94.434	58.550	66.706	58.327	119.976	130.648
@10	0.024	0.014	0.021	0.0	0.0	0.0	0.0	0.013
@30	0.161	0.198	0.165	0.0	0.0	0.0	0.0	0.05

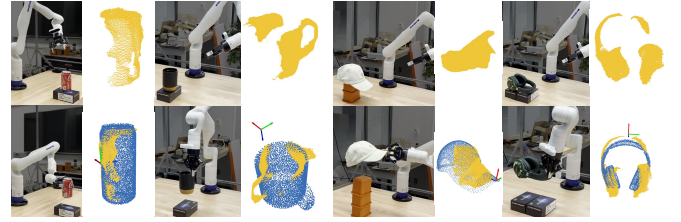


Fig. 4: Robotic manipulation experiment. By registering the yellow observed input to the blue template shape, the grasping pose defined in the template frame can be transformed into the camera frame such that the relative gripper pose can be obtained.

shape as the template from the Shapenet [63] dataset and predefine a grasping pose with respect to this template frame. We randomly initialize the manipulator pose such that the camera is roughly pointing at the target object. As shown in the top row of Fig. 4, given an observed partial point cloud from an unknown pose, we register the observation to the synthetic template shape with our method. In the bottom row of Fig. 4, we show that after estimating the transformation between the camera frame and the template frame, the relative transformation between the current gripper pose and the grasping pose can be naturally obtained (with the known camera to gripper transformation). Using the estimated relative gripper pose, we show that all four objects can be picked up. Despite the mismatching of the mug’s handle, we are still able to pick up the mug along the edge.

#### V. CONCLUSION, LIMITATIONS, AND FUTURE WORK

In this paper, we presented a novel optimization method for point cloud registration that considers one-to-many key-point correspondences together with a new feature descriptor. We showed that our proposed matching pipeline not only outperforms the SoTA but also that components of our method can be used as a drop-in replacement to improve existing methods. Furthermore, we showed that our method is less sensitive to noise and generalizes well on synthetic and real-world data compared to existing methods. Robotic manipulation experiments were also conducted to show that our method applies directly to real-world applications.

One limitation of our method is using FPS for keypoint extraction, a source of randomness. Although we showed that FPS is sufficient in our case for keypoint selection, there might be other settings where an additional keypoint detection module could help select more informative keypoints. Moreover, our LPS feature descriptor relies on ground normal estimation, which might not always be accurate. The local reference frame construction retains one degree of freedom ambiguity when the ground normal aligns with the point normal. Our method does not address the variations in scale. In future work, we plan to combine our approach with a learned feature descriptor to overcome these issues.

## REFERENCES

- [1] Haotian Zhang, Gaoang Wang, Zhichao Lei, and Jenq-Neng Hwang. Eye in the sky: Drone-based object tracking and 3d localization. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 899–907, 2019.
- [2] Nikolaus Correll, Kostas E Bekris, Dmitry Berenson, Oliver Brock, Albert Causo, Kris Hauser, Kei Okada, Alberto Rodriguez, Joseph M Romano, and Peter R Wurman. Analysis and observations from the first amazon picking challenge. *IEEE Transactions on Automation Science and Engineering*, 15(1):172–188, 2016.
- [3] Sara Zollini, Maria Alicandro, Donatella Dominici, Raimondo Quaresima, and Marco Giallonardo. Uav photogrammetry for concrete bridge inspection using object-based image analysis (obia). *Remote Sensing*, 12(19):3180, 2020.
- [4] David G Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [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. Spie, 1992.
- [6] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.
- [7] Radu Bogdan Rusu, Nico Blodow, Zoltan Csaba Marton, and Michael Beetz. Aligning point cloud views using persistent feature histograms. In *2008 IEEE/RSJ international conference on intelligent robots and systems*, pages 3384–3391. IEEE, 2008.
- [8] 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.
- [9] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Fast global registration. In *European conference on computer vision*, pages 766–782. Springer, 2016.
- [10] Max Schwarz, Hannes Schulz, and Sven Behnke. Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features. In *2015 IEEE international conference on robotics and automation (ICRA)*, pages 1329–1335. IEEE, 2015.
- [11] Alex Kendall, Matthew Grimes, and Roberto Cipolla. Posenet: A convolutional network for real-time 6-dof camera relocation. In *Proceedings of the IEEE international conference on computer vision*, pages 2938–2946, 2015.
- [12] Alex Kendall and Roberto Cipolla. Geometric loss functions for camera pose regression with deep learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5974–5983, 2017.
- [13] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.
- [14] Bugra Tekin, Sudipta N Sinha, and Pascal Fua. Real-time seamless single shot 6d object pose prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 292–301, 2018.
- [15] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. *arXiv preprint arXiv:1807.03146*, 2018.
- [16] Georgios Pavlakos, XiaoWei Zhou, Aaron Chan, Konstantinos G Derpanis, and Kostas Daniilidis. 6-dof object pose from semantic keypoints. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 2011–2018. IEEE, 2017.
- [17] Wentao Yuan, Benjamin Eckart, Kihwan Kim, Varun Jampani, Dieter Fox, and Jan Kautz. Deepgmr: Learning latent gaussian mixture models for registration. In *European conference on computer vision*, pages 733–750. Springer, 2020.
- [18] Dvir Ginzburg and Dan Raviv. Deep confidence guided distance for 3d partial shape registration. *arXiv preprint arXiv:2201.11379*, 2022.
- [19] Yue Wang and Justin M Solomon. Deep closest point: Learning representations for point cloud registration. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3523–3532, 2019.
- [20] Zi Jian Yew and Gim Hee Lee. Rpm-net: Robust point matching using learned features. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [21] Bingli Wu, Jie Ma, Gaojie Chen, and Pei An. Feature interactive representation for point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5530–5539, 2021.
- [22] Donghoon Lee, Onur C Hamsici, Steven Feng, Prachee Sharma, and Thorsten Gernoth. Deeppro: Deep partial point cloud registration of objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5683–5692, 2021.
- [23] Kexue Fu, Shaolei Liu, Xiaoyuan Luo, and Manning Wang. Robust point cloud registration framework based on deep graph matching. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [24] Yasuhiro Aoki, Hunter Goforth, Rangaprasad Arun Srivatsan, and Simon Lucey. Pointnetlk: Robust & efficient point cloud registration using pointnet. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [25] Szymon Rusinkiewicz and Marc Levoy. Efficient variants of the icp algorithm. In *Proceedings third international conference on 3-D digital imaging and modeling*, pages 145–152. IEEE, 2001.
- [26] François Pomerleau, Francis Colas, Roland Siegwart, et al. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015.
- [27] Wolfgang Forstner and Kourosh Khoshelham. Efficient and accurate registration of point clouds with plane to plane correspondences. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2165–2173, 2017.
- [28] H. Yang, J. Shi, and L. Carlone. TEASER: Fast and Certifiable Point Cloud Registration. *IEEE Trans. Robotics*, 2020.
- [29] Heng Yang and Luca Carlone. A polynomial-time solution for robust registration with extreme outlier rates. *arXiv preprint arXiv:1903.08588*, 2019.
- [30] 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.
- [31] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1802–1811, 2017.
- [32] 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/CVF conference on computer vision and pattern recognition*, pages 5545–5554, 2019.
- [33] 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.
- [34] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.
- [35] Yue Wang and Justin M Solomon. Pnet: Self-supervised learning for partial-to-partial registration. *Advances in neural information processing systems*, 32, 2019.
- [36] Haobo Jiang, Yaqi Shen, Jin Xie, Jun Li, Jianjun Qian, and Jian Yang. Sampling network guided cross-entropy method for unsupervised point cloud registration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6128–6137, 2021.
- [37] Jiahao Li, Changhao Zhang, Ziyao Xu, Hangning Zhou, and Chi Zhang. Iterative distance-aware similarity matrix convolution with mutual-supervised point elimination for efficient point cloud registration. In *European Conference on Computer Vision (ECCV)*, 2020.
- [38] Zheng Qin, Hao Yu, Changjian Wang, Yulan Guo, Yuxing Peng, and Kai Xu. Geometric transformer for fast and robust point cloud registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11143–11152, June 2022.
- [39] Shengyu Huang, Zan Gojcic, Mikhail Usvyatsov, and Konrad Schindler Andreas Wieser. Predator: Registration of 3d point clouds with low overlap. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2021.
- [40] 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.

- [41] Davis Rempe, Tolga Birdal, Yongheng Zhao, Zan Gojcic, Srinath Sridhar, and Leonidas J Guibas. Caspr: Learning canonical spatiotemporal point cloud representations. *Advances in neural information processing systems*, 33:13688–13701, 2020.
- [42] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3dpo: Canonical 3d pose networks for non-rigid structure from motion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7688–7697, 2019.
- [43] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019.
- [44] Riccardo Spezialetti, Federico Stella, Marlon Marcon, Luciano Silva, Samuele Salti, and Luigi Di Stefano. Learning to orient surfaces by self-supervised spherical cnns. *Advances in Neural Information Processing Systems*, 33:5381–5392, 2020.
- [45] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey Hinton, and Kwang Moo Yi. Canonical capsules: Self-supervised capsules in canonical pose. In *Neural Information Processing Systems*, 2021.
- [46] Xiaolong Li, Yijia Weng, Li Yi, Leonidas Guibas, A Lynn Abbott, Shuran Song, and He Wang. Leveraging se(3) equivariance for self-supervised category-level object pose estimation. *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.
- [47] Rahul Sajjani, Adrien Poulencard, Jivitesh Jain, Radhika Dua, Leonidas J. Guibas, and Srinath Sridhar. Condor: Self-supervised canonicalization of 3d pose for partial shapes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022.
- [48] Oren Katzir, Dani Lischinski, and Daniel Cohen-Or. Shape-pose disentanglement using se(3)-equivariant vector neurons. *arXiv preprint arXiv:2204.01159*, 2022.
- [49] Yosuke Nakagawa, Hideaki Uchiyama, Hajime Nagahara, and Rinchiro Taniguchi. Estimating surface normals with depth image gradients for fast and accurate registration. In *2015 International Conference on 3D Vision*, pages 640–647. IEEE, 2015.
- [50] Hugues Hoppe, Tony DeRose, Tom Duchamp, John McDonald, and Werner Stuetzle. Surface reconstruction from unorganized points. In *Proceedings of the 19th annual conference on computer graphics and interactive techniques*, pages 71–78, 1992.
- [51] Xufang Pang, Feng Li, Ning Ding, and Xiaopin Zhong. Upright-net: Learning upright orientation for 3d point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14911–14919, 2022.
- [52] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. A metric for distributions with applications to image databases. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 59–66. IEEE, 1998.
- [53] Maxim Tatarchenko\*, Stephan R. Richter\*, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? 2019.
- [54] Olga Sorkine-Hornung and Michael Rabinovich. Least-squares rigid motion using svd. *Computing*, 1(1):1–5, 2017.
- [55] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.
- [56] Tomas Hodan, Frank Michel, Eric Brachmann, Wadim Kehl, Anders GlentBuch, Dirk Kraft, Bertram Drost, Joel Vidal, Stephan Ihrke, Xenophon Zabulis, et al. Bop: Benchmark for 6d object pose estimation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 19–34, 2018.
- [57] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016.
- [58] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3d: A modern library for 3d data processing. *arXiv preprint arXiv:1801.09847*, 2018.
- [59] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020.
- [60] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.
- [61] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [62] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.
- [63] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.