CORSAIR: Convolutional Object Retrieval and Symmetry-AIded Registration

Tianyu Zhao Qiaojun Feng Sai Jadhav Nikolay Atanasov

Abstract— **This paper considers online object-level mapping using partial point-cloud observations obtained online in an unknown environment. We develop an approach for fully Convolutional Object Retrieval and Symmetry-AIded Registration (CORSAIR). Our model extends the Fully Convolutional Geometric Features model to learn a global object-shape embedding in addition to local point-wise features from the point-cloud observations. The global feature is used to retrieve a similar object from a category database, and the local features are used for robust pose registration between the observed and the retrieved object. Our formulation also leverages symmetries, present in the object shapes, to obtain promising local-feature pairs from different symmetry classes for matching. We present results from synthetic and real-world datasets with different object categories to verify the robustness of our method.**

I. Introduction

Advances in learning-based computer vision algorithms have enabled detection, classification, and segmentation of objects in images and videos with impressive accuracy. However, spatial and temporal perception of 3D environments at an object level using streaming sensory data remains a challenging task. It involves problems such as joint object pose and shape estimation from multi-view observations, data association of object instances across time and space, and compressed object shape representation for large-scale mapping. While general object reconstructions [1] without a prior object model is affected by occlusions, sensor noise, or segmentation algorithm misclassification, utilizing CAD object models to fit the noise observations may enable efficient and accurate object-level maps [2], [3].

Most existing work focuses on category-level object retrieval or object retrieval with fixed poses [4]–[6] or pose estimation between identical object pairs [7]–[9], differing only due to noise or occlusion but not due to shape variation. This paper considers object pose estimation from partial point-cloud observations to enable online object-level mapping. We assume that an object detection and segmentation model, trained offline using a large database of images or object CAD models, is available to segment object instances across multiple camera views and provide partial point-cloud observations. For each observed instance, we focus on retrieving a similar CAD model from the offline shape database and aligning it to the observation to estimate the observed object's pose. Fig. 1 illustrates the problem setting.

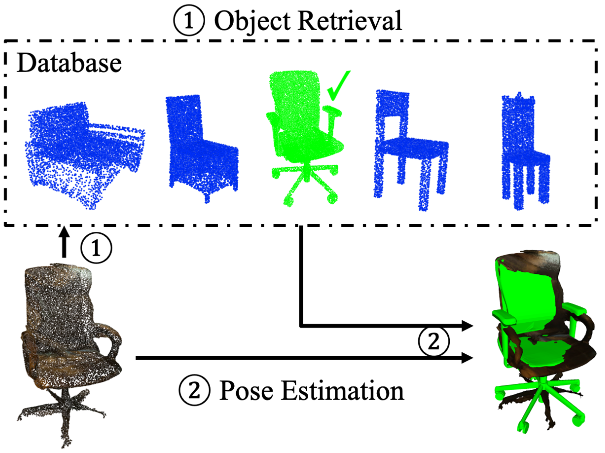

Fig. 1: Given an observed object point-cloud, we focus on retrieving a similar object (green) from a category database (blue) and estimating its pose with respect to the input object. Performing retrieval and registration for object point-clouds observed online allow us to construct an object-level map of an unknown environment.

Given a query point-cloud, we extract global features to enable retrieval and local point-wise features to enable point-cloud matching and pose registration. We present CORSAIR, an approach for fully convolutional object retrieval and symmetry-aided registration. CORSAIR extends the Fully Convolutional Geometric Features (FCGF) model [7] by introducing a latent code embedding architecture, which provides a global object-level feature in addition to the local point-wise features. The local point-wise features are trained using cross-object point matches in normalized canonical coordinates [10]. The global feature is extracted from a hierarchical abstraction of the bottleneck layer of the FCGF encoder-decoder architecture. Both are trained using metric learning with contrastive loss for the local features and triplet loss for the global feature. Our formulation also leverages symmetries in the object shapes to obtain promising feature matches in each symmetric class, greatly improving pose estimation. Our **contributions** are summarized as follows.

- We design a sparse fully convolutional network to jointly regress global and local point-cloud features, which are hierarchically correlated. The global feature enables similar model retrieval, while the local features allow object pose registration.
- We construct symmetry classes within an object instance based on the local features and aid the generation of promising feature pairs for robust registration.

The effectiveness of these contributions is validated in both synthetic and real-world object datasets, containing different

We gratefully acknowledge support from ARL DCIST CRA W911NF-17-2-0181 and ONR SAI N00014-18-1-2828.

The authors are with the Department of Electrical and Computer Engineering, University of California San Diego, La Jolla, CA 92093, USA {tiz007,qjfeng,smjadhav,natanasov}@ucsd.edu.

object categories.

II. Related Work

Deep neural network techniques for object detection and semantic segmentation [11], [12] have reached impressive levels of performance. They have allowed traditional dense geometric simultaneous localization and mapping (SLAM) approaches [13], [14] to integrate semantic content in the scene reconstruction [15], [16]. Since many robotics applications involve object interaction, SLAM algorithms which provide sparse object-level reconstruction [1], [2], [17], [18] instead of dense maps play an important role as well.

While object detection, segmentation and tracking are already well studied, we focus on object pose estimation from partial point-clouds obtained online for object-level mapping. To estimate poses of unknown instances, we rely on registration of known instances from the same category with respect to the observed point-cloud. The availability of massive object CAD datasets [19] makes it possible to select a similar-looking instance from the database to increase the accuracy of estimating the pose of the unknown instance. Hence, object retrieval is an important sub-problem for robust pose registration. Compared to category classification of 3D objects [4], [5], retrieval of specific CAD instances is more challenging due to the emphasis on shape similarity. Grabner et al. [20] render CAD model depth and embed the depth and RGB image observations jointly for CAD model retrieval. Dahnert et al. [21] use a 3D hourglass encoder-decoders structure to learn an embedding feature with triplet loss for shapes, implicitly represented using a signed distance field. Uy et al. [6] introduce a deformation-aware asymmetric distance across CAD models and learn an egocentric anisotropic distance field for latent embeddings. Most of these works, however, perform retrieval with canonical object poses and, hence, do not consider pose invariance or registration. On the other hand, many point-cloud registration approaches assume the point-clouds are from the same object or scene. DGR [8] predicts correspondence confidence of 3D point pairs using a 6D convolutional network and applies a weighted Procrustes algorithm, making the whole process differentiable. TEASER [9] is a certifiable registration algorithm handling high outlier rates with a truncated least squares formulation and semidefinite relaxation.

The Scan2CAD dataset [3] annotates 6D pose and scale of objects in the indoor scenes of ScanNet [22] by aligning CAD models from ShapeNet. RGBD scans are converted to voxelized signed distance fields and a 3D CNN network is used to predict sparse keypoint correspondence, given a matching CAD model. NOCS [10] uses the idea of normalized canonical coordinates for a specific category and generates dense annotations covering the whole object surface. This model can predict the normalized canonical coordinates densely on a query image and use them to recover the object pose. Feng et al. [23] align different object instances from the same category in normalized canonical coordinates and learn cross-instance matching FCGF features. Mask2CAD [24] detects and segments objects in a single RGB image, after which a CAD model is retrieved and its pose is regressed. The CAD model embedding is generated by rendering 2D images from different views to overcome the modality gap. Vid2CAD [25] leverages multi-view consistency constraints to resolve scale and depth ambiguities so as to derive a temporally consistent pose estimation of the objects.

We extend the work of [23], which tackles intra-category point-cloud matching by enabling retrieval of a similar instance from the category database for matching. We show that a global object-level feature for retrieval can be generated hierarchically from the local point-wise FCGF features for matching and alignment. We also leverage the symmetry of artificial objects to derive a more robust pose estimation approach.

III. Problem Formulation

Consider a robot, equipped with an RGBD camera, aiming to construct an object-level map of an unknown environment. Assume that the camera pose is estimated using an odometry algorithm, such as ORB-SLAM3 [26]. Assume also that a convolutional neural network, such as Mask R-CNN [11], is used to detect and segment objects in each RGB image, and an object tracking algorithm, such as FairMOT [27], tracks the object detections over time. A partial point-cloud observation $\mathbf{X} \in \mathbb{R}^{3\times N}$ of a tracked object instance can be obtained by accumulating the segmented RGBD pixels associated with the instance over time and projecting them to the world frame using the estimated camera pose trajectory.

Let $\mathcal{Y} := \left\{\mathbf{Y}_i \in \mathbb{R}^{3\times M_i}\right\}_i$ be a database of point-cloud object models from the same category as $\mathbf{X}$. We assume the database was used offline for training the object detection and tracking models and is available to the robot. We consider the following joint object retrieval and registration problem.

Problem 1. Given a query point-cloud $\mathbf{X} \in \mathbb{R}^{3\times N}$ and a point-cloud database $\mathcal{Y} := \left\{\mathbf{Y}_i \in \mathbb{R}^{3\times M_i}\right\}_i$, retrieve a point-cloud $\mathbf{Y} \in \mathcal{Y}$ that is similar to $\mathbf{X}$ and estimate its rotation $\mathbf{R} \in SO(3)$ and translation $\mathbf{p} \in \mathbb{R}^3$ with respect to $\mathbf{X}$:

$$\min_{\mathbf{Y}\in\mathcal{Y},\mathbf{R}\in SO(3),\mathbf{p}\in\mathbb{R}^3} d(\mathbf{X}, \mathbf{R}\mathbf{Y} + \mathbf{p}\mathbf{1}^\top), \tag{1}$$

where $\mathbf{1}$ is a vector with all elements equal to 1 and d is a point-cloud distance metric.

There are different ways to specify a point-cloud distance d in Problem 1. We measure the average distance between matching points $\mathbf{x}_i \in \mathbb{R}^3$ in $\mathbf{X}$ and $\mathbf{y}_{m(i)} \in \mathbb{R}^3$ in $\mathbf{Y}$:

$$d(\mathbf{X}, \mathbf{Y}) = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{y}_{m(i)} - \mathbf{x}_i\|^2 \mathbb{1}_{\{m(i)\neq 0\}}, \tag{2}$$

where $m : \{1, \dots N\} \mapsto \{0, 1, \dots, M\}$ associates the indices i of the points in $\mathbf{X}$ with the indices $j = m(i)$ of the points in $\mathbf{Y}$, and $m(i) = 0$ indicates that i does not match any index in $\mathbf{Y}$.

The objective of Problem 1 is to determine the world-frame pose of an object instance, observed online, which may or may not have been seen before. Retrieving a similar instance from the training database and registering it with

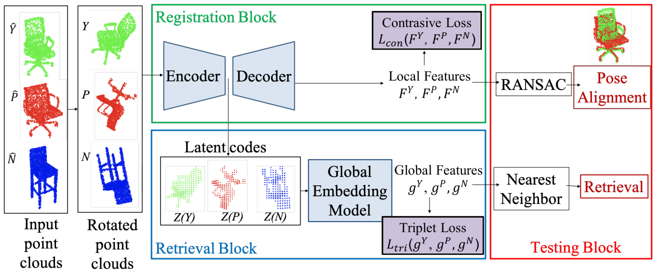

Fig. 2: During training, given a point-cloud and its corresponding positive and negative pairs $(\mathbf{Y}, \mathbf{P}, \mathbf{N})$, a Registration block is trained to generate local point-wise features (Sec. IV-A) and a Retrieval block is trained to generate a global shape embedding (Sec. IV-B). During testing (Sec. IV-C), given a query point-cloud $\mathbf{X}$, we generate its global embedding $\mathbf{g}^{\mathbf{X}}$ and retrieve a similar instance using nearest neighbors in the embedding space. Local features are then generated for both point-clouds, and matching pairs are used to recover the pose of the query using RANSAC.

the point-cloud observation allows accurate pose estimation of the newly observed object.

IV. Approach

Estimating the pose of novel objects based on a finite set of models from the same category can be challenging due to shape variation. We extend the point-wise FCGF feature extractor proposed in [7] with a global embedding network. We learn *local point-wise features* (Sec. IV-A) to enable robust matching and registration of point-cloud with potentially different shapes. We learn *global object-level features* (Sec. IV-B) to enable retrieval of a point cloud from the database that is similar to the query point cloud. During inference (Sec. IV-C), we align a partially observed point-cloud with a retrieved one, and exploit object symmetry to generate matching feature pairs for registration. Fig. 2 presents an overview of our approach.

A. Local Features for Pose Registration

We aim to predict matching pairs of point-wise local feature for pose registration between point clouds. During training, we first define matching pairs of points, rather than features. If two point-clouds $\mathbf{X} \in \mathbb{R}^{3\times N}$ and $\mathbf{Y} \in \mathbb{R}^{3\times M}$ were already aligned in the same coordinate frame, matching point pairs can be extracted via:

$$p(\mathbf{X}, \mathbf{Y}) = \{(i,j) \in \mathbb{N}^2 \mid \|\mathbf{x}_i - \mathbf{y}_j\| < \tau, i \leq N, j \leq M\}, \quad (3)$$

where $\tau > 0$ is a matching tolerance. Negative pairs can be obtained from the complement set of the positive pairs, ensuring that two negatively associated points are at least a margin τ away.

Since the training set $\mathcal{Y}$ contains point clouds from different instances, which inherently carry geometric shape differences, we generate matching pairs in category-level normalized canonical coordinates (NCCs) [10], [23]. However, instead of dense correspondence annotation as done in [10], [23], we only need object pose annotations to convert the point-clouds to NCCs. Given the scale $s_{\mathbf{X}} \in \mathbb{R}$, rotation $\mathbf{R}_{\mathbf{X}} \in SO(3)$, and translation $\mathbf{p}_{\mathbf{X}} \in \mathbb{R}^3$ of a point-cloud $\mathbf{X}$ during training, $\mathbf{X}$ can be converted to NCCs via:

$$\mathbf{X}_{\text{NCC}} = s_{\mathbf{X}}^{-1} \cdot \mathbf{R}_{\mathbf{X}}^{\top} \left(\mathbf{X} - \mathbf{p}_{\mathbf{X}} \mathbf{1}^{\top}\right). \quad (4)$$

Thus, to generate matching pairs for different instances $\mathbf{X}$ and $\mathbf{Y}$ of the same category, we convert both into NCCs, and obtain the positive pair set as $\mathcal{P}_{\mathcal{L}} = p(\mathbf{X}_{\text{NCC}}, \mathbf{Y}_{\text{NCC}})$. A negative pair set $\mathcal{N}_{\mathcal{L}}$ is obtained as a subset of the complement of $\mathcal{P}_{\mathcal{L}}$.

Given a point-cloud $\mathbf{X} \in \mathbb{R}^{3\times N}$, our model uses a sparse fully convolutional encoder-decoder architecture, illustrated in Fig. 3, to predict local point-wise features:

$$\mathbf{F}^{\mathbf{X}} = [\mathbf{f}_1^{\mathbf{x}}, \ldots, \mathbf{f}_N^{\mathbf{x}}] \in \mathbb{R}^{C\times N}, \quad (5)$$

where $\mathbf{f}_i^{\mathbf{x}}$ is the feature corresponding to the point $\mathbf{x}_i$. Sparse convolution [28] generalizes image convolution to arbitrary dimensions and coordinates and allows processing of spatially sparse inputs. Our model is an extension of the FCGF model [7] that adds an embedding module to the encoder output (bottleneck layer) to also retrieve a global feature. The training and role of the global feature for retrieval is described in Sec. IV-B.

We use metric learning to train the local feature extractor. Relying on the positive pairs $\mathcal{P}_{\mathcal{L}}$ and the negative pairs $\mathcal{N}_{\mathcal{L}}$ of matching points, we define a contrastive loss function for the features $\mathbf{F}^{\mathbf{X}}$ and $\mathbf{F}^{\mathbf{Y}}$ associated with two point clouds from the training set:

$$\begin{aligned} L_{\text{con}}(\mathbf{F}^{\mathbf{X}}, \mathbf{F}^{\mathbf{Y}}) = &\sum_{(i,j)\in\mathcal{P}_{\mathcal{L}}} \max(0, \|\mathbf{f}_i^{\mathbf{x}} - \mathbf{f}_j^{\mathbf{y}}\|_2 - p_+)^2 \\ &+ \sum_{(i,j)\in\mathcal{N}_{\mathcal{L}}} \max(0, p_- - \|\mathbf{f}_i^{\mathbf{x}} - \mathbf{f}_j^{\mathbf{y}}\|_2)^2, \end{aligned} \quad (6)$$

where p_+ and p_- are the positive and negative thresholds. These thresholds are selected to ensure that points from the positive pairs move closer together and points from the negative pairs move farther apart in the feature space. We normalize the feature vectors to unit length and set $p_+ = 0.1$ and $p_- = 1.5$.

For each point cloud $\mathbf{Y}$ in the training set $\mathcal{Y}$, we choose similar point clouds $\mathbf{P}$ and dissimilar point clouds $\mathbf{N}$ defined precisely in Sec. IV-B. We, then, generate the positive pairs between $\mathbf{Y}$ and $\mathbf{P}$ using (3) as $\mathcal{P}_{\mathcal{L}} = p(\mathbf{Y}_{NCC}, \mathbf{P}_{NCC})$. We sample the negative pair set $\mathcal{N}_{\mathcal{L}}$ by taking random pairs between $\mathbf{Y}$ and $\mathbf{N}$ as well as from the complement of $\mathcal{P}_{\mathcal{L}}$. The local feature extractor model is trained with the contrastive loss in (6) but using the pairs from $\mathcal{P}_{\mathcal{L}}$ and $\mathcal{N}_{\mathcal{L}}$.

B. Global Feature for Object Retrieval

Extracting a similar point cloud from $\mathcal{Y}$ for a given query point cloud $\mathbf{X}$ is crucial for pose registration because only similar shapes provide consistent local geometric features for matching. Since the sparse convolutional model in Fig. 3 performs in providing local geometric features, we design a global shape descriptor by combining the local features. An input point cloud $\mathbf{X}$ is quantized into a sparse tensor and gets downsampled by the encoder into a point cloud $\mathbf{Z}(\mathbf{X}) \in$

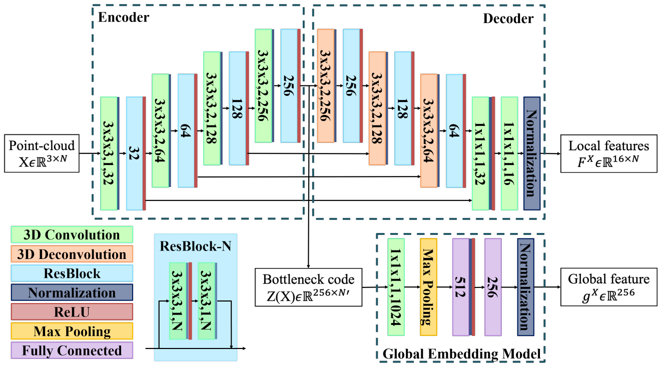

Fig. 3: Our model extends the sparse convolutional encoder-decoder ResUNet structure proposed in FCGF [7] by adding an embedding module to the bottleneck latent code (encoder output). The output of our embedding module provides a global object shape feature, suitable for retrieval, while the decoder generates point-wise local features. The numbers in the 3D convolution and deconvolution blocks represent kernel sizes, strides, and output dimensions. The numbers in the fully connected blocks represent the output dimensions. Each ResBlock is composed of two 3D convolution layers and the number indicates the output dimensions.

$\mathbb{R}^{256\times N'}$ with fewer points $N' < N$ but each in a higher (256 in our case) dimension. After the multiple convolution layers of the encoder, the bottleneck code $\mathbf{Z}(\mathbf{X})$ encodes a high-level structure feature which should be beneficial for shape retrieval. We introduce an embedding module to extract a single global feature $\mathbf{g}^{\mathbf{X}} \in \mathbb{R}^{256}$ from $\mathbf{Z}(\mathbf{X})$ as $\mathbf{g}^{\mathbf{X}} = g(\mathbf{Z}(\mathbf{X}))$. As shown in Fig. 3, the embedding module $g(.)$, includes a fully convolutional layer, followed by maxpooling to combine the features from all points in $\mathbf{Z}(\mathbf{X})$ and pass them through several fully connected layers to obtain $\mathbf{g}^{\mathbf{X}}$.

We use metric learning for global feature training too. To measure the similarity of point cloud $\mathbf{X} \in \mathbb{R}^{3\times N}$ with respect to $\mathbf{Y} \in \mathbb{R}^{3\times M}$, we define a Single-direction Chamfer Distance (SCD):

$$d_{SCD}\left(\mathbf{X}, \mathbf{Y}\right) = \sum_{i=1}^{N} \min_{j=1}^{M} \|\mathbf{x}_i - \mathbf{y}_j\|_2^2. \tag{7}$$

The bi-directional similarity between the two point clouds is measured by the usual Chamfer distance:

$$d_{CD}\left(\mathbf{X}, \mathbf{Y}\right) = d_{SCD}\left(\mathbf{X}, \mathbf{Y}\right) + d_{SCD}\left(\mathbf{Y}, \mathbf{X}\right) \tag{8}$$

Let $\mathbf{D} \in \mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$ encode the pair-wise Chamfer distance similarity of all point clouds in $\mathcal{Y}$ with elements:

$$\mathbf{D}_{i,j} = d_{CD}(\mathbf{Y}_i, \mathbf{Y}_j), \;\; \text{for } \mathbf{Y}_i, \mathbf{Y}_j \in \mathcal{Y}. \tag{9}$$

A similarity ranking for $\mathbf{Y}_i \in \mathcal{Y}$ can be obtained by sorting the i-th column of $\mathbf{D}$ in ascending order. We define a positive set $\mathcal{P}_{\mathcal{G}}(\mathbf{Y}_i)$ and negative set $\mathcal{N}_{\mathcal{G}}(\mathbf{Y}_i)$ of point clouds associated with $\mathbf{Y}_i$ as follows:

$$\mathcal{P}_{\mathcal{G}}(\mathbf{Y}_i) = \{\mathbf{Y}_j \mid \text{Rank}_i(\mathbf{Y}_j) \le \tau_+|\mathcal{Y}|, d_{CD}(\mathbf{Y}_i, \mathbf{Y}_j) \le \delta_+\}$$
$$\mathcal{N}_{\mathcal{G}}(\mathbf{Y}_i) = \{\mathbf{Y}_j \mid \text{Rank}_i(\mathbf{Y}_j) \ge \tau_-|\mathcal{Y}|, d_{CD}(\mathbf{Y}_i, \mathbf{Y}_j) \ge \delta_-\}$$

where $\text{Rank}_i(\mathbf{Y})$ returns an integer indicating the Chamfer distance ranking of $\mathbf{Y}$ to $\mathbf{Y}_i$, τ_+ and τ_- are the percentage

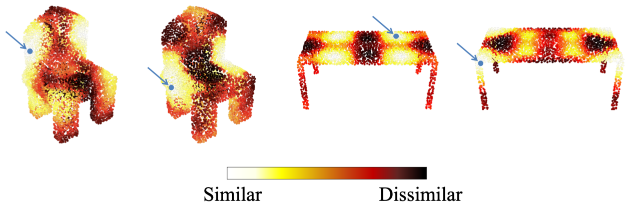

Fig. 4: Heatmap of local feature similarity for different points on a chair (left) and a table (right) instances. Lighter points indicate points with similar features to the query point (blue dot), while darker ones have dissimilar features. The feature similarity heatmap visualizes the symmetric nature of the local features.

of positive and negative point clouds we want to consider, and δ_+ and δ_- are the Chamfer distance thresholds. In our experiments, we set $\tau_+ = 0.1$, $\tau_- = 0.5$, $\delta_+ = 0.15$ and $\delta_- = 0.20$.

For a given point-cloud $\mathbf{Y}$, we randomly select one positive object $\mathbf{P} \in \mathcal{P}_{\mathcal{G}}(\mathbf{Y})$ and one negative object $\mathbf{N} \in \mathcal{N}_{\mathcal{G}}(\mathbf{Y})$ and train the global embedding module with a triplet loss:

$$L_{\text{tri}}(\mathbf{g}^{\mathbf{Y}}, \mathbf{g}^{\mathbf{P}}, \mathbf{g}^{\mathbf{N}}) = \max(1 + \|\mathbf{g}^{\mathbf{Y}} - \mathbf{g}^{\mathbf{P}}\|_2 - \|\mathbf{g}^{\mathbf{Y}} - \mathbf{g}^{\mathbf{N}}\|_2, 0). \tag{10}$$

Similar to the contrastive loss for the local features, the triplet loss pushes similar point-clouds closer and drags dissimilar point-clouds apart in the global embedding space.

C. Inference

Finally, we consider pose registration for a query point cloud $\mathbf{X}$ given the trained CORSAIR model in Fig. 3. The first step is to retrieve a similar model from $\mathcal{Y}$. The local and global features of all point clouds in $\mathcal{Y}$ are pre-computed offline. The query $\mathbf{X}$ is passed through the CORSAIR model to obtain its local features $\mathbf{F}^{\mathbf{X}}$ and global feature $\mathbf{g}^{\mathbf{X}}$. The instance from $\mathbf{Y}$ closest to $\mathbf{X}$ is retrieved via:

$$\mathbf{Y} = \mathop{\arg\min}_{\mathbf{Y_i} \in \mathcal{Y}} \|\mathbf{g}^{\mathbf{X}} - \mathbf{g}^{\mathbf{Y_i}}\|_2, \tag{11}$$

Next, we generate matching pairs between $\mathbf{X}$ and $\mathbf{Y}$ by searching for K nearest neighbors in the local feature space:

$$nn_K(\mathbf{F}^{\mathbf{X}}, \mathbf{F}^{\mathbf{Y}}) = \{(i, j_i) | j_i \in K\text{-}\mathop{\arg\min}_{j} \|\mathbf{f}_i^{\mathbf{x}} - \mathbf{f}_j^{\mathbf{y}}\|_2, \quad i \le N, j_i \le M, i, j_i \in \mathbb{N}\} \tag{12}$$

where we define $K\text{-}\arg\min_j$ as the set of K different values that make the function smaller than any other set of K indices. The correspondence candidates in $nn_K(\mathbf{F}^{\mathbf{X}}, \mathbf{F}^{\mathbf{Y}})$ are used to recover the rotation and translation of $\mathbf{Y}$ with respect to $\mathbf{X}$ via a robust pose estimation method such as RANSAC [29].

Artificial objects usually have one or more planes of symmetry. Since our model is able to generate rotation-invariant local features, the features from symmetrical areas can be similar, which significantly increases the risk of mismatch in (12). A feature-distance heatmap shown in Fig. 4 illustrates the symmetrical patterns. We propose a symmetry-aware method to add constraints in the nearest neighbor matching phase. We split the point-cloud with a symmetric segmentation method (see Alg. 1) and then

Algorithm 1 Symmetry-aided Segmentation

```
input: point-cloud $\mathbf{X}$, point-wise local features $\mathbf{F}^{\mathbf{X}}$, number of matching feature pairs $M$, number of symmetry classes $G$
Randomly sample $\mathcal{S} \leftarrow \{\mathbf{x}_i | \mathbf{x}_i \in \mathbf{X}, i \leq n\}$
for $\mathbf{x}_i \in \mathcal{S}$ do
    $\mathcal{K} \leftarrow \{\mathbf{x}_j | (i,j) \in nn_K(\mathbf{f}_i^{\mathbf{X}}, \mathbf{F}^{\mathbf{X}})\}$
    Split $\mathbf{X}$ in clusters $\mathcal{C}_i = \{\widetilde{\mathbf{X}}_1, \ldots, \widetilde{\mathbf{X}}_G\}$ via GMEANS($\mathcal{M}$)
    $\sigma_i \leftarrow$ Standard Deviation of $\{|\widetilde{\mathbf{X}}_1|, \ldots, |\widetilde{\mathbf{X}}_G|\}$
output: $\mathcal{C}_i$ with the smallest $\sigma_i$
```

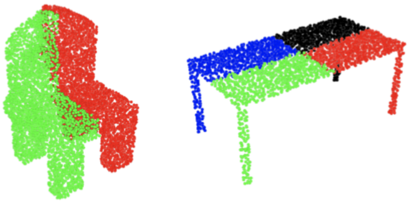

Fig. 5: Different objects split according to their symmetry planes.

constrain the nearest-neighbor matching to the corresponding parts.

Given an object category, we assume that the number of symmetric classes G, computed as the number of symmetry planes times 2, is known. For example, $G = 2$ for chairs and $G = 4$ for tables. We first extract the point-wise local features $\mathbf{F}^{\mathbf{X}}$ for the input point-cloud $\mathbf{X}$. Second, we randomly sample n points from the point-cloud. For each sampled point, we take its K nearest neighbors, $nn_K(\mathbf{f}_i^{\mathbf{X}}, \mathbf{F}^{\mathbf{X}})$, and perform G-means clustering using their 3D spatial coordinates. The object can then be split into G parts, $\{\widetilde{\mathbf{X}}_1, \ldots, \widetilde{\mathbf{X}}_G\}$, by the decision boundaries of G-means clustering as shown in Fig. 5. Each of the splits is considered as a candidate and we choose the most even split as our symmetric segmentation output. The evenness of a split is measured by the standard deviation σ of the sizes of the G parts.

We assume that the query and retrieved point clouds $\mathbf{X}$ and $\mathbf{Y}$ share the same symmetry property and split them with our symmetry segmentation method. Since there are multiple possible mappings between the subsets $\{\widetilde{\mathbf{X}}_\mathbf{1}, \ldots, \widetilde{\mathbf{X}}_G\}$ and $\{\widetilde{\mathbf{Y}}_\mathbf{1}, \ldots, \widetilde{\mathbf{Y}}_G\}$, we generate matching pairs using (12) for all the possible mappings. We also generate matching pairs without the symmetry constraints as a back-up for asymmetric objects. We supply these sets of matching pairs to RANSAC to estimate the rotation $\mathbf{R}$ and translation $\mathbf{p}$ that align $\mathbf{Y}$ with $\mathbf{X}$ for every possible matching pair set. The quality of alignment is evaluated by the single direction Chamfer distance $d_{SCD}(\mathbf{X}, \mathbf{RY} + \mathbf{p1}^\top)$ defined in (7). The rotation and translation with the best alignment quality are selected as the output of our symmetry-aware pose estimation method. Our symmetry-aware method performs well when the input point-clouds have symmetrical structure.

V. Evaluation

We evaluate the performance of CORSAIR on the synthetic ShapeNet [19] dataset and the real-world Scan2CAD [3] dataset. We assume that the category of any given point-cloud is known so that we can train models for different categories separately. In our experiments, chair and table are selected.

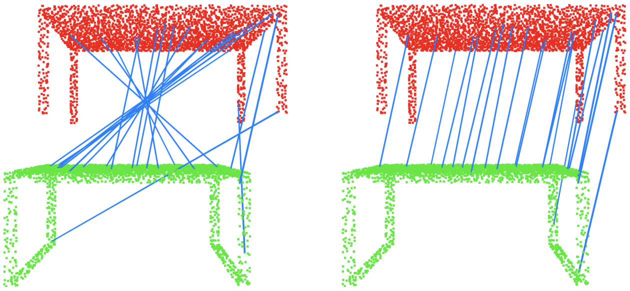

Fig. 6: Local feature matching between a query point-cloud (red) and a retrieved point-cloud (green). The blue lines show the pairs between the two point-clouds based on nearest neighbor matching of the local features. The left pair of point-clouds shows matching without symmetry aid, while the right pair shows matching after detecting the plane of symmetry.

A. Evaluation Metrics

Given a query point-cloud $\mathbf{X}$ and its retrieved point-cloud $\mathbf{Y}$, we define the ground truth pose of $\mathbf{X}$ with respect to $\mathbf{Y}$ as $(\mathbf{R}^*, \mathbf{p}^*)$ and the estimated pose as $(\hat{\mathbf{R}}, \hat{\mathbf{p}})$. We compute relative rotation error (RRE) and relative translation error (RTE) as pose estimation metrics:

$$\mathbf{RTE}(\hat{\mathbf{p}}, \mathbf{p}^*) = ||\hat{\mathbf{p}} - \mathbf{p}^*||_2, \tag{13}$$

$$\mathbf{RRE}(\hat{\mathbf{R}}, \mathbf{R}^*) = \arccos((\mathbf{tr}(\hat{\mathbf{R}}^\top \mathbf{R}^*) - 1)/2). \tag{14}$$

Two different metrics are used to evaluate the retrieval performance on ShapeNet: Precision@M and Top-1 Chamfer distance. Precision@M is defined as:

$$\text{Precision@M} = \frac{1}{M} \sum_{\mathbf{Y} \in \mathcal{R}} \mathbb{1}_{\mathbf{Y} \in \mathcal{P}_\mathcal{G}(\mathbf{X})}, \tag{15}$$

where $\mathcal{R} \subset \mathcal{Y}$ is the retrieved set of M CAD models and $\mathcal{P}_\mathcal{G}(\mathbf{X})$ is the ground truth positive set for query object $\mathbf{X}$ as defined in IV-B. The Top-1 Chamfer distance metric is calculated by measuring the Chamfer distance (8) between the top-1 retrieved CAD model and the ground-truth top-1 similar CAD model.

B. ShapeNet

The ShapeNet dataset [19] contains a wide range of CAD models with category labels. We use point-clouds sampled from the CAD models provided by [31]. In the category of chair, we use 4612, 592, 1242 point-clouds for training, validation and testing. In the category of table, we use 5744, 778, 1557 point-clouds for training, validation and testing. The shape retrieval and pose estimation tasks are performed within each category. We first train the local feature extractor with positive and negative matching pairs as mentioned in eq. (3) for 100 epochs. Then we freeze the parameters and train the embedding network with triplets $(\mathbf{Y}, \mathbf{P}, \mathbf{N})$ for another 100 epochs. Random rotation is applied to each of $\mathbf{Y}$, $\mathbf{P}$ and $\mathbf{N}$ before training and evaluating. The random seed is fixed in all the experiments for fair comparisons.

In ShapeNet, pose registration and the retrieval are considered as two separate tasks. To evaluate the pose registration

TABLE I: Quantitative pose registration results in ShapeNet [19]. Every object is aligned with a randomly selected object from its top-10% similar positive set. RANSAC is applied for all the methods for pose registration. The table shows the percentage of test cases below different error thresholds for different feature extraction and matching strategies.

Category	Method	RRE≤ 5°	RRE≤ 15°	RRE≤ 45°	RTE≤ 0.03	RTE≤ 0.05	RTE≤ 0.10
Chair	FPFH [30]	1.4	7.8	18.5	2.0	10.7	25.8
	FCGF [7]	41.0	86.1	96.1	32.2	83.8	96.0
	CORSAIR (Ours)	62.3	92.1	98.1	48.5	90.2	97.7
Table	FPFH [30]	3.8	18.2	34.4	3.0	14.9	32.5
	FCGF [7]	25.4	62.7	78.4	24.9	63.8	80.5
	CORSAIR (Ours)	54.5	74.8	82.7	46.0	77.0	85.0

TABLE II: Ablation evaluation of CORSAIR in Scan2CAD [3]. We compare our top-1 retrieved CAD (Top-1 Retrieval) and the ground-truth CAD annotation (GT annotation). We also compare our symmetry-aware matching method (w/ sym) with naive nearest neighbor (w/o sym). The percentage of test cases lower than different thresholds and the average single-direction Chamfer distance are reported.

Category	CAD model	Registration	RRE ≤ 5°	RRE ≤ 15°	RRE ≤ 45°	RTE≤ 0.05	RTE≤ 0.10	RTE≤ 0.15	$SCD(\times 10^{-2})$
Chair	GT Annotation	w/o sym	25.2	82.2	89.9	21.6	60.4	78.7	5.99
		w/ sym	34.4	87.9	94.0	27.4	68.5	85.2	5.43
	Top-1 Retrieval	w/o sym	15.4	72.5	86.4	8.2	35.7	59.0	7.53
		w/ sym	23.2	78.5	88.9	9.7	40.5	63.5	6.81
Table	GT Annotation	w/o sym	19.2	58.4	72.5	12.7	38.5	58.8	7.20
		w/ sym	32.3	69.4	76.3	26.1	56.7	70.1	5.68
	Top-1 Retrieval	w/o sym	11.7	40.9	52.9	4.8	17.5	28.5	9.06
		w/ sym	24.5	50.2	57.0	10.7	27.8	41.2	7.14

TABLE III: Retrieval quantitative results in ShapeNet [19].

Method	Chair		Table	
	Precision@M	Top-1 CD	Precision@M	Top-1 CD
3D ResNet18 [32]	21.81	0.182	17.49	0.231
PointNet [33]	25.65	0.188	17.76	0.234
FCGF [7]	31.83	0.132	36.19	0.135
CORSAIR (Ours)	51.47	0.115	57.77	0.112

performance, we estimate the transformation between a CAD model $\mathbf{Y}_i$ and a similar object in $\mathbf{Y}_j \in \mathcal{P}_G(\mathbf{Y}_i)$ and measure the RRE and RTE. We assume that the symmetry of the $\mathbf{Y}_j$ is known and $\mathbf{Y}_i$ shares the same symmetry. In the evaluation of retrieval, we use the metric defined in V-A. We report Precision@M$= 0.1n$, where n is the size of the test set.

In the pose registration task, we compare our learned local feature (based on FCGF) with the hand-crafted FPFH feature [30]. FCGF is the same local feature extractor as ours but it generates matching pairs without our symmetry-aware method. RANSAC is applied to estimate the pose with given matching pairs in (12) with $K = 5$. Fig. 6 is a qualitative result showing that our method generates more accurate matching pairs with the aid of symmetry information than the naive nearest-neighbor method. Mismatches caused by symmetry areas are filtered out. The quantitative results are shown in Table I. Our symmetry-aware method outperforms FCGF baseline by 21.3% and 29.1% for chairs and tables, in terms of the ratio of test cases with RRE $\leq 5°$. The results show that our symmetry-aware method improves the pose estimation performance by refining matching pairs.

We compare our retrieval module with other prevalent global shape descriptors including 3D ResNet18 [32] and PointNet [33] as well as FCGF without our global embedding network. We use the 3D ResNet18 implementation in [28]. Both 3D ResNet18, PointNet and our method generate a 256-D global descriptor for retrieval. The FCGF method directly uses latent vector $\mathbf{Z}(\mathbf{X})$ as the global feature, and the distance is measured by the Chamfer distance (8) in 256-D. All the methods are trained with the same loss function as defined in (10). Quantitative results are shown in Table III. Our method outperforms the baseline by a large margin in both Precision@M and Top-1 Chamfer distance metrics.

C. Scan2CAD

The Scan2CAD dataset [3] provides object-level human-generated annotations. The annotations includes category label, segmentation, a similar CAD model in ShapeNet, and the corresponding pose. In this dataset, we assume that the segmentation and category are known but the annotated similar CAD model and the pose are unknown. We use the object segmentation labels to segment the object meshes from the scene and sample points on the surfaces to convert them to point-clouds. In the chair category, we use 2896, 343, 993 scanned point-clouds for training, validation and testing. In the table category, we use 1164, 150, 291 scanned point-clouds for training, validation and testing. Since the object distribution in the scenes is not uniform, we split the dataset by scenes instead of objects. Given pretrained parameters from ShapeNet, we train the local feature extractor and the embedding network on the Scan2CAD dataset separately for 100 epochs each. In the training phase, random rotations are applied to both scanned objects and CAD models. In the evaluation phase, we set the CAD models in the canonical pose.

For the pose registration task, we assume the number of symmetry classes for the CAD models are known in advance, and the scanned objects share the same symmetry with retrieved CAD models. For the retrieval task, the database $\mathcal{Y}$ contains CAD models from the Scan2CAD dataset and belong to the same category as the scanned object. The size of the CAD model database is 652 and 830 for the chair and table categories, respectively. Unlike in the ShapeNet experiments, for a scanned object $\mathbf{X}$, we only consider the ground-truth annotated CAD model as positive object $\mathbf{P}$. The negative object $\mathbf{N}$ is randomly sampled from the negative set $\mathcal{N}_G(\mathbf{P})$ with respect to the annotated positive object $\mathbf{P}$, since the similarity between partially scanned objects and a CAD

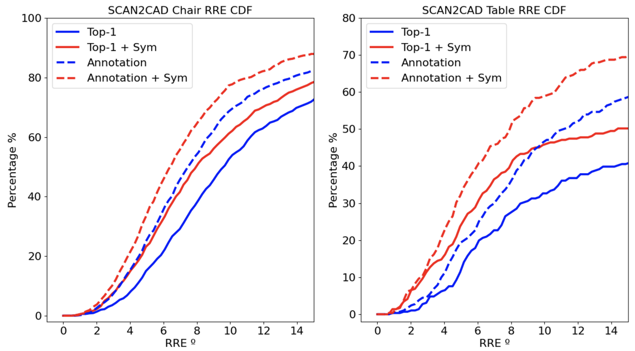

Fig. 7: Cumulative distribution function of RRE on the Scan2CAD dataset [3]. The solid lines represent RRE when aligning with top-1 retrieved CAD models. The dotted lines represents RRE aligning with annotated CAD models. Blue lines use naive nearest neighbor features + RANSAC for registration, while the red lines use our symmetry-aided nearest neighbor matching + RANSAC.

model is not well-defined. The retrieval task is evaluated jointly with the pose estimation task. We report the RRE and RTE as well as the single direction Chamfer distance to assess the overall alignment quality.

We first evaluate the pose registration method by aligning scanned objects with the annotated CAD models. See Table II, Fig. 7 for details. In a real world setting, our pose registration method is still able to estimate accurate poses (RRE $\leq 5°$) for 25.2% of the chairs and 19.2% of the tables. The error for the majority of the test cases is within a reasonable range (RRE $\leq 15°$). With our symmetry-aware method, we can further improve the ratio of accurate estimation by 9.2% and 13.1% for chairs and tables. The symmetry-aware method still generates better pose results in both categories when point-clouds are partially observed. Our method works well on approximately complete point-clouds, and for severely occluded scans we use the naive nearest neighbors with RANSAC as a back-up to handle asymmetric cases. Then we evaluate both the retrieval and the pose estimation using single direction Chamfer distance in the last column of Table II. Our retrieved CAD models can reach comparable alignment results compared with human-labeled ground-truth CAD models. This indicate that our method is able to retrieve reasonable models for better alignment.

In Fig. 8, we visualize the scene-level reconstruction to show qualitative results in real-world scenarios. Our method is able to retrieve CAD models that share the same structure with the scanned objects and align them accurately. Some failed cases are also presented, e.g., the chair on the right of the second scene. Most of the failed cases are caused by severe occlusions. The absence of key structures, like the legs of a table or chair, may lead to multiple solutions. The limitations of raw point-cloud measurements make it hard for our method to solve this kind of problems.

VI. Conclusion

This work proposed CORSAIR, an approach for category-level retrieval and registration. CORSAIR is a fully convolutional model for point-cloud processing which jointly generates local point-wise geometric features and a global rotation-invariant shape feature. The global feature allows retrieval of similar object instances from the same category, while the local features, aided by symmetry class labeling, provide matching pairs for pose registration between the retrieved and query objects. For retrieval, CORSAIR outperforms the baseline methods by a large margin and even achieves comparable results when compared with human annotations. The symmetry-aware method proposed in CORSAIR refines the matching pair based on the naive nearest neighbor method and leads to considerable improvement on pose registration. Currently the global and local features extraction takes 0.2s for each object. While the retrieval step can run at 300 Hz, the registration using RANSAC runs slower than 1 Hz. We will explore faster alternatives for robust registration. Future work will also focus on making the pose estimation stage differentiable as well to enable end-to-end training of the whole model, using only pose annotations.

References

[1] M. Runz, M. Buffier, and L. Agapito, "MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2018, pp. 10–20.

[2] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas, "Probabilistic data association for semantic SLAM," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 1722–1729.

[3] A. Avetisyan, M. Dahnert, A. Dai, M. Savva, A. X. Chang, and M. Nießner, "Scan2CAD: Learning CAD Model Alignment in RGB-D Scans," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 2609–2618.

[4] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning SO(3) Equivariant Representations with Spherical CNNs," in *Computer Vision – ECCV*, 2018, pp. 54–70.

[5] X. He, Y. Zhou, Z. Zhou, S. Bai, and X. Bai, "Triplet-Center Loss for Multi-view 3D Object Retrieval," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1945–1954.

[6] M. A. Uy, J. Huang, M. Sung, T. Birdal, and L. Guibas, "Deformation-Aware 3D Model Embedding and Retrieval," in *Computer Vision – ECCV*, 2020, pp. 397–413.

[7] C. Choy, J. Park, and V. Koltun, "Fully Convolutional Geometric Features," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8957–8965.

[8] C. Choy, W. Dong, and V. Koltun, "Deep Global Registration," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2511–2520.

[9] H. Yang, J. Shi, and L. Carlone, "TEASER: Fast and Certifiable Point Cloud Registration," *IEEE Transactions on Robotics*, pp. 1–20, 2020.

[10] H. Wang, S. Sridhar, J. Huang, J. Valentin, S. Song, and L. J. Guibas, "Normalized Object Coordinate Space for Category-Level 6D Object Pose and Size Estimation," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019, pp. 2637–2646.

[11] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 386–397, Feb 2020.

[12] S. H. Gao, M. M. Cheng, K. Zhao, X. Y. Zhang, M. H. Yang, and P. Torr, "Res2Net: A New Multi-Scale Backbone Architecture," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 2, pp. 652–662, 2021.

[13] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger, "ElasticFusion: Real-time dense SLAM and light source estimation," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1697–1716, 2016.

[14] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-Time Globally Consistent 3D Reconstruction Using On-the-Fly Surface Reintegration," *ACM Trans. Graph.*, vol. 36, no. 3, May 2017.

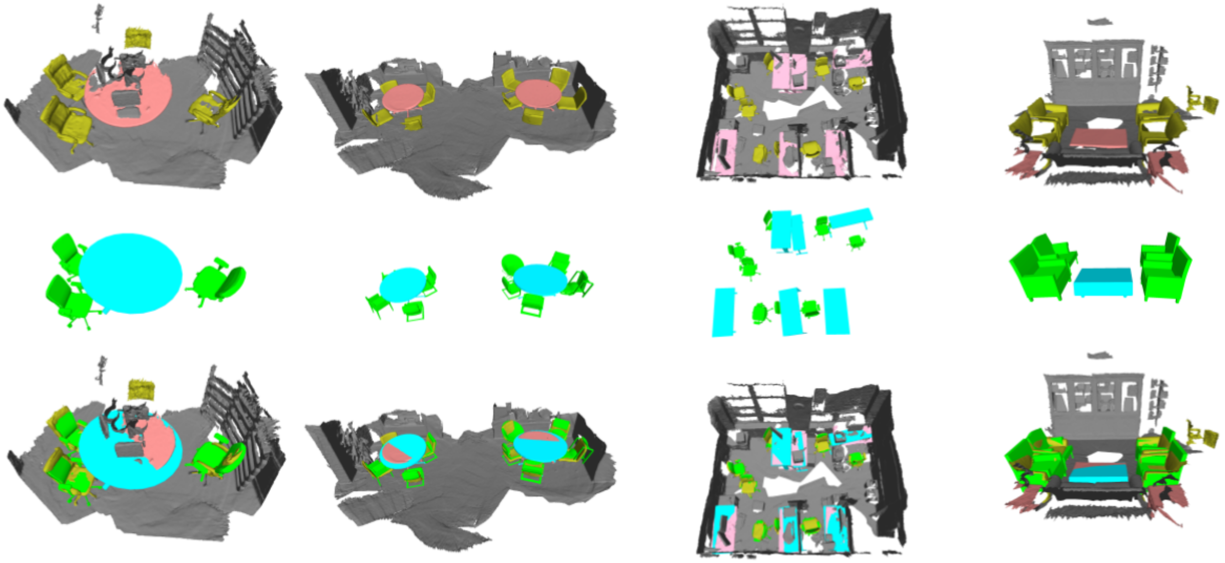

Fig. 8: Scene-level reconstruction of scenes 0314_00, 0355_00, 0653_00 and 0690_00 from the Scan2CAD dataset [3]. The segmented chairs (yellow) and tables (pink) in the first row are inputs to our model. The second row shows the predicted object map, obtained from aligning retrieved instances to the query point-clouds using CORSAIR. The retrieved chairs (green) and tables (blue) are overlaid back into the scene to visualize the reconstruction qualitatively.

[15] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6565–6574.

[16] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4628–4635.

[17] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly, and A. J. Davison, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013, pp. 1352–1359.

[18] J. Mccormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric Object-Level SLAM," in *International Conference on 3D Vision (3DV)*, 2018, pp. 32–41.

[19] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," *ArXiv*, vol. abs/1512.03012, 2015.

[20] A. Grabner, P. M. Roth, and V. Lepetit, "3D Pose Estimation and 3D Model Retrieval for Objects in the Wild," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3022–3031.

[21] M. Dahnert, A. Dai, L. Guibas, and M. Niessner, "Joint Embedding of 3D Scan and CAD Objects," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 8748–8757.

[22] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-Annotated 3D Reconstructions of Indoor Scenes," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2432–2443.

[23] Q. Feng and N. Atanasov, "Fully Convolutional Geometric Features for Category-level Object Alignment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8492–8498.

[24] W. Kuo, A. Angelova, T.-Y. Lin, and A. Dai, "Mask2CAD: 3D Shape Prediction by Learning to Segment and Retrieve," in *Computer Vision – ECCV*, 2020, pp. 260–277.

[25] K. Maninis, S. Popov, M. Nießner, and V. Ferrari, "Vid2CAD: CAD Model Alignment using Multi-View Constraints from Videos," *ArXiv*, vol. abs/2012.04641, 2020.

[26] C. Campos, R. Elvira, J. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial and Multi-Map SLAM," *ArXiv*, vol. abs/2007.11898, 2020.

[27] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking," *arXiv*, vol. abs/2004.01888, 2020.

[28] C. Choy, J. Gwak, and S. Savarese, "4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 3070–3079.

[29] M. A. Fischler and R. C. Bolles, *Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography*, 1987, p. 726–740.

[30] R. B. Rusu, N. Blodow, and M. Beetz, "Fast Point Feature Histograms (FPFH) for 3D registration," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[31] G. Yang, X. Huang, Z. Hao, M. Liu, S. Belongie, and B. Hariharan, "PointFlow: 3D Point Cloud Generation With Continuous Normalizing Flows," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 4540–4549.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[33] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 77–85.