End-to-End CAD Model Retrieval and 9DoF Alignment in 3D Scans

Armen Avetisyan Angela Dai Matthias Nießner
Technical University of Munich

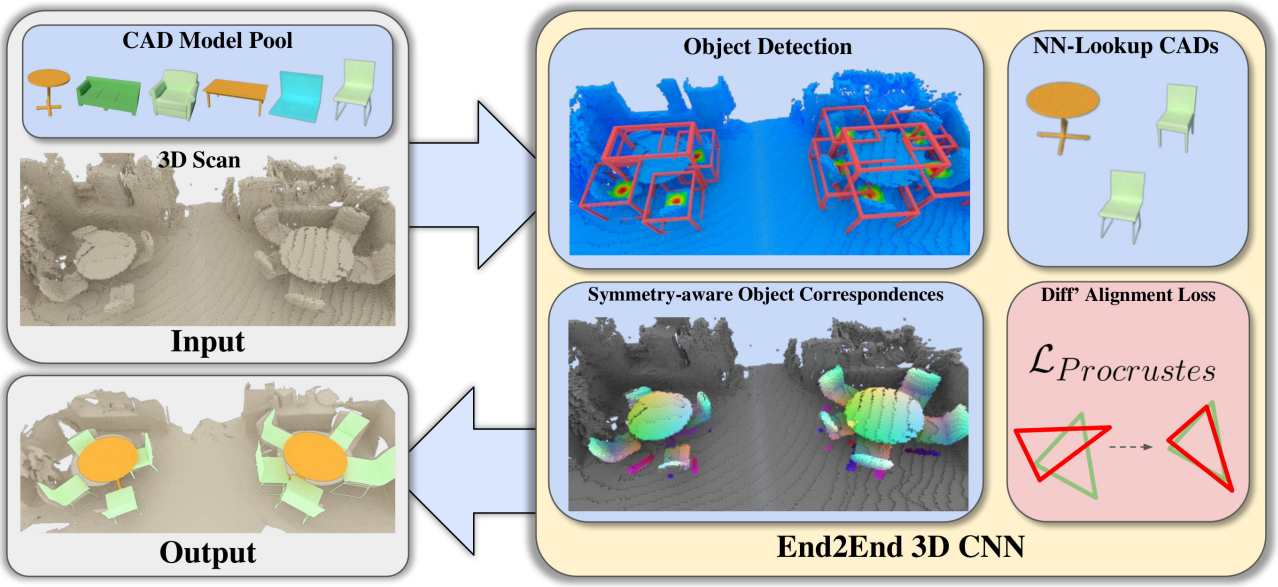

Figure 1: From a 3D scan and a set of CAD models, our method learns to predict 9DoF CAD model alignments to the objects of the scan in a fully-convolutional, end-to-end fashion. Our proposed 3D CNN first detects objects in the scan, then uses the regressed object bounding boxes to establish symmetry-aware object correspondences between a scan object and CAD model, which inform our differentiable Procrustes alignment loss, enabling learning of alignment-informed correspondences and producing CAD model alignment to a scan in a single forward pass.

Abstract

We present a novel, end-to-end approach to align CAD models to an 3D scan of a scene, enabling transformation of a noisy, incomplete 3D scan to a compact, CAD reconstruction with clean, complete object geometry. Our main contribution lies in formulating a differentiable Procrustes alignment that is paired with a symmetry-aware dense object correspondence prediction. To simultaneously align CAD models to all the objects of a scanned scene, our approach detects object locations, then predicts symmetry-aware dense object correspondences between scan and CAD geometry in a unified object space, as well as a nearest neighbor CAD model, both of which are then used to inform a differentiable Procrustes alignment. Our approach operates in a fully-convolutional fashion, enabling alignment of CAD models to the objects of a scan in a single forward pass. This enables our method to outperform state-of-the-art approaches by 19.04% *for CAD model alignment to scans, with* $\approx 250\times$ *faster runtime than previous data-driven approaches.*

1. Introduction

In recent years, RGB-D scanning and reconstruction has seen significant advances, driven by the increasing availability of commodity range sensors such as the Microsoft Kinect, Intel RealSense, or Google Tango. State-of-the-art 3D reconstruction approaches can now achieve impressive capture and reconstruction of real-world environments [19, 26, 27, 38, 38, 5, 8], spurring forth many potential applications of this digitization, such as content creation, or augmented or virtual reality.

Such advances in 3D scan reconstruction have nonetheless remained limited towards these use scenarios, due to geometric incompleteness, noise and oversmoothing, and lack of fine-scale sharp detail. In particular, there is a notable contrast in such reconstructed scan geometry in comparison to the clean, sharp 3D models created by artists for visual and graphics applications.

With the increasing availability of synthetic CAD models [4], we have the opportunity to reconstruct a 3D scan

through CAD model shape primitives; that is, finding and aligning similar CAD models from a database to each object in a scan. Such a scan-to-CAD transformation enables construction of a clean, compact representation of a scene, more akin to artist-created 3D models to be consumed by mixed reality or design applications. Here, a key challenge lies in finding and aligning similar CAD models to scanned objects, due to strong low-level differences between CAD model geometry (clean, complete) and scan geometry (noisy, incomplete). Current approaches towards this problem thus often operate in a sparse correspondence-based fashion [22, 1] in order to establish reasonable robustness under such differences.

Unfortunately, such approaches, in order to find and align CAD models to an input scan, thus involve several independent steps of correspondence finding, correspondence matching, and finally an optimization over potential matching correspondences for each candidate CAD model. With such decoupled steps, there is a lack of feedback through the pipeline; e.g., correspondences can be learned, but they are not informed by the final alignment task. In contrast, we propose to predict symmetry-aware dense object correspondences between scan and CADs in a global fashion. For an input scan, we leverage a fully-convolutional 3D neural network to first detect object locations, and then from each object location predict a uniform set of dense object correspondences and object symmetry are predicted, along with a nearest neighbor CAD model; from these, we introduce a differentiable Procrustes alignment, producing a final set of CAD models and 9DoF alignments to the scan in an end-to-end fashion. Our approach outperforms state-of-the-art methods for CAD model alignment by 19.04% for real-world 3D scans.

Our approach is the first, to the best of our knowledge, to present an end-to-end scan-to-CAD alignment, constructing a CAD model reconstruction of a scene in a single forward pass. In summary, we propose an end-to-end approach for scan-to-CAD alignment featuring:

- a novel differentiable Procrustes alignment loss, enabling end-to-end CAD model alignment to a 3D scan,
- symmetry-aware dense object correspondence prediction, enabling robust alignment even under various object symmetries, and
- CAD model alignment for a scan of a scene in a single forward pass, enabling very efficient runtime ($<$ 3s on real-world scan evaluation)

2. Related work

RGB-D Scanning and Reconstruction 3D scanning methods have a long research history across several communities, ranging from offline to real-time techniques. In particular, RGB-D scanning has become increasingly popular, due to the increasing availability of commodity range sensors. A very popular reconstruction technique is the volumetric fusion approach by Curless and Levoy [6], which has been materialized in many real-time reconstruction frameworks such as KinectFusion [19, 26], Voxel Hashing [27] or BundleFusion [8], as well as in the context of state-of-the-art offline reconstruction methods [5]. An alternative to these voxel-based scene representations is based on surfels [21], which has been used by ElasticFusion [38] to realize loop closure updates. These works have led to RGB-D scanning methods that feature robust, global tracking and can capture very large 3D environments. However, though these methods can achieve stunning results in RGB-D capture and tracking, the quality of reconstructed 3D geometry nonetheless remains far from from artist-created 3D content, as the reconstructed scans are partial, and contain noise or oversmoothing from sensor quality or small camera tracking errors.

3D Features for Shape Alignment and Retrieval An alternative to bottom-up 3D reconstruction from RGB-D scanning techniques is to find high-quality CAD models that can replace the noisy and incomplete geometry from a 3D scan. Finding and aligning these CAD models inevitably requires 3D feature descriptors to find geometric matches between the scan and the CAD models. Traditionally, these descriptors were hand-crafted, and often based on a computation of histograms (e.g., point normals), such as FPFH [30], SHOT [36], or point-pair features [12].

More recently, with advances in deep neural networks, these descriptors can be learned, for instance based on an implicit signed distance field representation [41, 10, 11]. A typical pipeline for CAD-to-scan alignments builds on these descriptors; i.e., the first step is to find 3D feature matches and then use a variant of RANSAC or PnP to compute 6DoF or 9Dof CAD alignments. This two-step strategy has been used by Slam++ [31], Li et al. [22], Shao et al. [32], the data-driven work by Nan et al. [25] and the recent Scan2CAD approach [1]. One potential approach to combine correspondence prediction and alignment is through differentiable RANSAC [3], which has been applied for camera localization. Our approach is designed to learn robust dense correspondences through a differentiable Procrustes alignment where correspondences and their relative weights are jointly optimized together without requiring multiple hypothesis generation. Other approaches rely only on single RGB(-D) frame input, but use a similar two-step alignment strategy [23, 20, 34, 18, 13, 42]. While these methods are related, their focus is different as we address geometric alignment independent of RGB information.

While promising results have been achieved by these two-step approaches, there remains a fundamental limita-

tion in the decoupled nature of feature matching and alignment computation. This inherently limits the ability of data-driven descriptors, as they remain unaware of the used optimization algorithm.

In our work, we propose an end-to-end alignment algorithm where correspondences are trained through gradients from an differentiable Procrustes optimizer.

Shape Retrieval Challenges and RGB-D Datasets In the context of 2D object alignment methods several datasets provide alignment annotations between RGB images and CAD models, including the PASCAL 3D+ [40], ObjectNet3D [39], the IKEA objects [23], and Pix3D [34]; however, no geometric information is given in the query images. SHREC provides a very popular series of 3D shape retrieval challenges, organized as part of Eurographics 3DOR [17, 29]; the tasks include matching objects from ScanNet [7] and SceneNN [16] to ShapeNet models [4].

More recently, Scan2CAD [1] provides accurate CAD alignment annotations on top of ScanNet [7] using ShapeNet models [4], based on roughly 100k manually annotated correspondences. In addition to evaluating our method on the Scan2CAD test dataset, we also evaluate on the synthetic SUNCG [33] dataset.

3. Overview

For an input 3D scan along with a set of candidate CAD models, our method aims to align similar CAD models to each object instance in the scan. Object locations in the scan are detected, and for each detected object, a similar CAD model is retrieved and a 9DoF transformation (3 degrees each for translation, rotation, and scale) computed to align it to the scan geometry. Thus we can transform a noisy, incomplete 3D scan into a compact, CAD-based representation with clean, complete geometry, as shown in Figure 1.

To this end, we propose an end-to-end 3D CNN-based approach to simultaneously retrieve and align CAD models to the objects of a scan in a single pass, for scans of varying sizes. This end-to-end formulation enables the final alignment process to inform learning of scan-CAD correspondences. To enable effective learning of scan-CAD object correspondences, we propose to use *symmetry-aware object correspondences* (SOCs), which establish dense correspondences between scan objects and CAD models, and are trained by our differentiable Procrustes alignment loss.

Then for an input scan $\mathbb{S}$ represented by volumetric grid encoding a truncated signed distance field, our model first detects object center locations as heatmap predictions over the volumetric grid and corresponding bounding box sizes for each object location. The bounding box represents the extent of the underlying object. From these detected object locations, we use the estimated bounding box size to crop out the neighborhood region around the object center from the learned feature space in order to predict our SOC correspondences to CAD models.

From this neighborhood of feature information, we then predict SOCs. These densely establish correspondences for each voxel in the object neighborhood to CAD model space. In order to be invariant to potential reflection and rotational symmetries, which could induce ambiguity in the correspondences, we simultaneously estimate the symmetry type of the object. We additionally predict a binary mask to segment the object instance from background clutter in the neighborhood, thus informing the set of correspondences to be used for the final alignment. To find a CAD model corresponding to the scan object, we jointly learn an object descriptor which is used to retrieve a semantically similar CAD model from a database.

Finally, we introduce a differentiable Procrustes alignment, enabling a fully end-to-end formulation, where learned scan object-CAD SOC correspondences can be informed by the final alignment process, achieving efficient and accurate 9DoF CAD model alignment for 3D scans.

4. Method

4.1. Network Architecture

Our network architecture is shown in Figure 2. It is designed to operate on 3D scans of varying sizes, in a fully-convolutional manner. An input scan is given by a volumetric grid encoding a truncated signed distance field, representing the scan geometry. We design our network backbone to learn features for detecting objects in a scan, establishing SOCs, and aligning CAD models to them. The end-to-end formulation enables the learned SOCs to be informed by the alignment performance.

The network backbone is structured in an encoder-decoder fashion, and composed of a series of ResNet blocks [14]. The bottleneck volume is spatially reduced by a factor of 16 from the input volume, and is decoded to the original resolution through transpose convolutions. The decoder is structured symmetrically to the encoder, but with half the feature channels, which we empirically found to produce faster convergence and more accurate performance. The output of the decoder is used to predict an objectness heatmap, identifying potential object locations, which is employed to inform bounding box regression for object detection. The predicted object bounding boxes are used to crop and extract features from the output of the second decoder layer, which then inform the SOC predictions. The features used to inform the SOC correspondence are extracted from the second block of the decoder, whose feature map spatial dimensions are $1/4$ of the original input dimension.

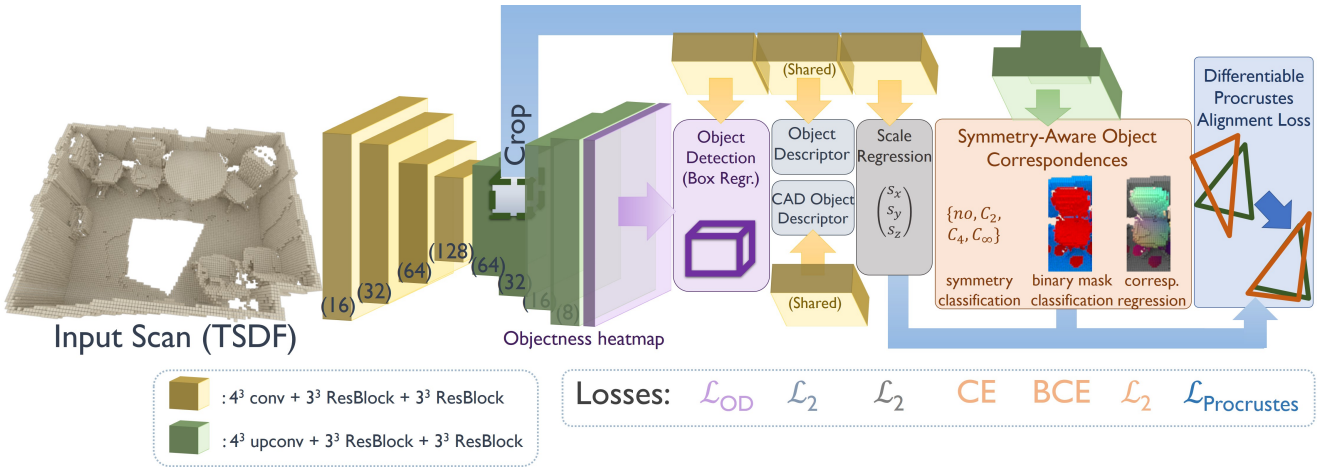

Figure 2: Network architecture for our end-to-end approach for CAD model alignment. An input TSDF scan represented in a volumetric grid is input to an encoder-decoder backbone constructed with residual blocks. Objects are detected through objectness prediction and bounding box regression; these predicted object boxes are then used to crop features from the decoder to inform CAD model alignment to a detected object. The cropped features are processed to simultaneously predict an object descriptor constrained to be similar to a corresponding CAD object descriptor (used for retrieving CAD models) and a 3-dimensional scale. Our symmetry-aware object correspondences (SOCs) informs directly our differentiable Procrustes alignment loss.

Object Detection We first detect objects, predicting bounding boxes for the objects in a scan, which then inform the SOC predictions. The output of the backbone decoder predicts heatmaps representing objectness probability over the full volumetric grid (whether a voxel is a center of an object). We then regress object bounding boxes corresponding to these potential object centers. For object bounding boxes predictions, we regress a 3-channel feature map, with each 3-dimensional vector corresponding to the bounding box extent size, and regressed using an ℓ_2 loss.

Objectness is predicted as a heatmap, encoding voxel-wise probabilities as to whether each voxel is a center of an object. Note that $\Omega \subset \mathbb{N}^3$ is the discretized space (i.e. voxel grid). To predict a location heatmap H_1, we additionally employ two proxy losses, using a second heatmap prediction H_2 as well as a predicted offset field O. H_1 and H_2 are two 1-channel heatmaps designed to encourage high recall and precision, respectively, and O is a 3-channel grid representing an offset field to the nearest object center. The objectness heatmap loss is:

$$\mathcal{L}_{OD} = 2.0 \cdot \mathcal{L}_{\text{recall}} + 10.0 \cdot \mathcal{L}_{\text{precision}} + 10.0 \cdot \mathcal{L}_{\text{offset}}$$

The weights for each component in the loss are designed to bring the losses numerically to approximately the same order of magnitude. Here, $\mathcal{L}_{\text{recall}}$ and $\mathcal{L}_{\text{precision}}$ are inspired from the conditional keypoint correspondence heatmap predictions of Scan2CAD [1].

$\mathcal{L}_{\text{recall}}$ aims to achieve high recall. It operates on the prediction H_1, on which we apply a sigmoid and calculate the loss via binary-cross entropy (BCE). This loss on its own tends to establish a high recall, but also blurry predictions.

$$\mathcal{L}_{\text{recall}} = \sum_{x \in \Omega} \text{BCE}(\sigma(H_1(x)), H_{\text{GT}}(x)) \quad (1)$$

$$H_1 : \Omega \rightarrow [0, 1], \quad \sigma : \text{sigmoid} \quad (2)$$

$\mathcal{L}_{\text{precision}}$ aims to achieve high precision. It operates on the prediction H_2, on which we apply a softmax and calculate the loss via negative log-likelihood (NLL). Due to the softmax, this loss encourages highly localized predictions in the output volume, which helps to attain high precision.

$$\mathcal{L}_{\text{precision}} = \sum_{x \in \Omega} \text{NLL}(\sigma(H_2(x)), H_{\text{GT}}(x)) \quad (3)$$

$$H_2 : \Omega \rightarrow [0, 1], \quad \sigma : \text{softmax} \quad (4)$$

$\mathcal{L}_{\text{offset}}$ is a regression loss on the predicted 3D offset field O, following [28]. Each voxel of O represents a 3-dimensional vector that points to the nearest object center. This regression loss is used as a proxy loss to support the other two classification losses.

$$\mathcal{L}_{\text{offset}} = \sum_{x \in \Omega} \|O(x) - O_{\text{GT}}(x)\|_2^2 \quad (5)$$

$$O : \Omega \rightarrow \mathbb{R}^3$$

Predicting SOCs SOCs are dense, voxel-wise correspondences from scan geometry to CAD models. They are de-

fined as SOC : $\Omega \rightarrow [-0.5, 0.5]^3$, the normalized space of the CAD models.

In order to account for symmetry ambiguities, ground truth SOCs are generated such that the front-facing axis of the CAD model maintains minimal angle with the x-axis of the scan voxel grid. Thus for symmetric objects, the SOCs are generated in a consistent fashion, i.e., always aligned with the x-axis of the scan coordinate system.

SOCs are predicted using features cropped from the network backbone. For each detected object, we crop a region with the extend of the predicted bounding box volume $\mathcal{F}$ from the feature map of the second upsampling layer to inform our dense, symmetry-aware object correspondences. This feature volume $\mathcal{F}$ is first fitted through tri-linear interpolation into a uniform voxel grid of size 48^3 before streaming into different prediction heads. SOCs incorporate several output predictions: a volume of dense correspondences from scan space to CAD object space, an instance segmentation mask, and a symmetry classification.

The dense correspondences, which map to CAD object space, implicitly contain CAD model alignment information. These correspondences are regressed as CAD object space coordinates, similar to [37], with the CAD object space defined as a uniform grid centered around the object, with coordinates normalized to $[-0.5, 0.5]$. These coordinates are regressed using an ℓ_2 loss.

We also introduce a proxy symmetry loss to encourage correct SOC prediction by predicting the symmetry class of the object for common symmetry classes for furniture objects: two-fold rotational symmetry, four-fold rotational symmetry, infinite rotational symmetry, and no symmetry.

Retrieval To retrieve a similar CAD model to the detected object, we use the cropped feature neighborhood $\mathcal{F}$ to train an object descriptor for the scan region, using a series of 3D convolutions to reduce the feature dimensionality. This resulting 512-dimensional object descriptor is then constrained to match the latent vector of an autoencoder trained on the CAD model dataset, with latent spaces constrained by an ℓ_2 loss. This enables retrieval of a semantically similar CAD model at test time through a nearest neighbor search using the object descriptor.

Scale Similarly to the retrieval head, the scale is predicted per detected object (i.e. per crop). We regress the $\mathbb{R}^3$ scale vector with an ℓ_2 loss. At train and test time this estimate is used as final scale estimate with further post processing.

9DoF Alignment Our differentiable 9DoF alignment enables training for CAD model alignment in an end-to-end fashion, thereby informing learned correspondences of the final alignment objective. To this end, we leverage a differentiable Procrustes loss on the masked correspondences given by the SOC predictions to find the rotation alignment. That is, we aim to find a rotation matrix R which brings together the CAD and scan correspondence points P_c, P_s:

$$R^* = \text{argmin}_R ||RP_c - P_s||_F, \qquad R \in SO_3$$

This is solved through a differentiable SVD of $P_s P_c^T = U\Sigma V^T$, with $R = U \begin{bmatrix} 1 & & \\ & 1 & \\ & & d \end{bmatrix} V^T$, $d = \det(VU^T)$. Here, the SVD is computed by solving the non-linear characteristic polynomial of the 3×3 matrix $P_s P_c^T$ iteratively, giving the final rotation. For scale and translation, we directly regress the scale using two 3D downsampling convolutions on $\mathcal{F}$, and the translation is predicted from the detected object centers. Note that an object center is the geometric center of the bounding box.

4.2. Training

Data Input scan data is represented by its truncated signed distance field (TSDF) encoded in a volumetric grid and generated through volumetric fusion [6] (we use voxel size = 3cm, truncation = 15cm). The CAD models used to train the autoencoder to produce a latent space for scan object descriptor training are represented as unsigned distance fields (DF), using the level-set generation toolkit by Batty [2].

To train our model for CAD model alignment for real scan data, we use the Scan2CAD dataset introduced by [1]. These Scan2CAD annotations provide 1506 scenes for training. Using upright rotation augmentation, we augment the number of training samples by 4 (90° increments with 20° random jitter). We train our network using full scenes as input, with batch size of 1. For SOC prediction at train time the batch size is equal to the number of groundtruth objects in the given scene as crops are only performed around groundtruth object centers. Only large scenes during training are randomly cropped to $400 \times 400 \times 64$ to meet memory requirements. We found that training using 1 scene per batch generally yields stable convergence behavior.

For CAD model alignment to synthetic scan data, we use the SUNCG dataset [33], where we virtually scan the scenes following [9, 15] to produce input partial TSDF scans. The training process for synthetic SUNCG scan data is identical to training with real data. See supplemental material for further details.

Optimization We use an SGD optimizer with a batch size of 1 scene and an initial learning rate of 0.002, which is decayed by 0.5 every $20K$ iterations. We train for $50K$ iterations until convergence, which takes ≈ 48 hours.

We train our model from scratch with the exception of the object retrieval descriptors. For object retrieval, we pretrain an autoencoder on all ShapeNetCore CAD models, trained to reconstruct their distance fields at 32^3. This CAD autoencoder is trained with a batch size of 16 for $30K$ iterations. We then train the full model with pre-trained object

	bath	bookshelf	cabinet	chair	display	sofa	table	trash bin	other	class avg.	avg.
FPFH (Rusu et al. [30])	0.00	1.92	0.00	10.00	0.00	5.41	2.04	1.75	2.00	2.57	4.45
SHOT (Tombari et al. [35])	0.00	1.43	1.16	7.08	0.59	3.57	1.47	0.44	0.75	1.83	3.14
Li et al. [22]	0.85	0.95	1.17	14.08	0.59	6.25	2.95	1.32	1.50	3.30	6.03
3DMatch (Zeng et al. [41])	0.00	5.67	2.86	21.25	2.41	10.91	6.98	3.62	4.65	6.48	10.29
Scan2CAD (Avetisyan et al. [1])	36.20	36.40	34.00	44.26	17.89	**70.63**	30.66	30.11	20.60	35.64	31.68
Direct 9DoF	5.88	13.89	13.48	21.94	2.78	8.04	10.53	13.01	17.65	11.91	15.12
Ours (no symmetry)	11.11	29.27	29.29	68.26	20.41	16.26	41.03	40.12	14.29	30	40.51
Ours (no SOCs)	11.11	21.95	7.07	61.77	8.16	9.76	28.21	17.9	19.48	20.6	29.97
Ours (no anchor)	**45.24**	**45.85**	47.16	61.55	**27.65**	51.92	41.21	31.13	**29.62**	42.37	47.64
Ours (no Procrustes)	33.33	36.59	28.28	50.51	14.29	13.01	58.97	35.19	28.57	33.19	35.74
Ours (final)	38.89	41.46	**51.52**	**73.04**	26.53	26.83	**76.92**	**48.15**	18.18	**44.61**	**50.72**

Table 1: Accuracy comparison (%) on Scan2CAD [1]. We compare to state-of-the-art handcrafted feature descriptors (FPFH [30], SHOT [35], Li et al. [22]) as well as learned descriptors (3DMatch [41], Scan2CAD [1]) for CAD model alignment. These approaches consider correspondence finding and pose alignment optimization independently, while our end-to-end formulation can learn correspondences informed by alignment, achieving significantly higher CAD model alignment accuracy.

Scene size	small	medium	large
Scene dim	$128 \times 96 \times 48$	$144 \times 128 \times 64$	$256 \times 320 \times 64$
# objects	7	16	20
Scan2CAD [1]	288.60s	565.86s	740.34s
Ours	**0.62s**	**1.11s**	**2.60s**

Table 2: Runtime (seconds) of our approach on varying-sized scenes. Our end-to-end approach predicts CAD model alignment in a single forward pass, enabling very efficient CAD model alignment – several hundred times faster than previous data-driven approaches.

descriptors for all ShapeNet models for CAD model alignment, with the CAD autoencoder latent space constraining the object descriptor training for retrieval.

5. Results

We evaluate our proposed end-to-end approach for CAD model alignment in comparison to the state of the art as well as with an ablation study analyzing our differentiable Procrustes alignment loss and various design choices. We evaluate on real-world scans using the Scan2CAD dataset [1]. We use the evaluation metric proposed by Scan2CAD [1]; that is, the ground truth CAD model pool is available as input, and a CAD model alignment is considered to be successful if the category of the CAD model matches that of the scan object and the alignment falls within 20cm, 20°, and 20% for translation, rotation, and scale, respectively. For further evaluation on synthetic scans, we refer to the supplemental material.

In addition to evaluating CAD model alignment using the Scan2CAD [1] evaluation metrics, we also evaluate our approach on an unconstrained scenario with 3000 random CAD models as a candidate pool, shown in Figure 4. In this scenario, we maintain robust CAD model alignment accuracy with a much larger set of possible CAD models.

Comparison to state of the art. Table 1 evaluates our approach against several state-of-the-art methods for CAD model alignment, which establish correspondences and alignment independently of each other. In particular, we compare to several approaches leveraging handcrafted feature descriptors: FPFH [30], SHOT [36], Li et al. [22], as well as learned feature descriptors: 3DMatch [41], Scan2CAD [1]. We follow these descriptors with RANSAC to obtain final alignment estimation, except for Scan2CAD, where we use the proposed alignment optimization. Our end-to-end formulation, where correspondence learning can be informed by the alignment, outperforms these decoupled approaches by over 19.04%. Figure 3 shows qualitative visualizations of our approach in comparison to these methods.

How much does the differentiable Procrustes alignment loss help? We additionally analyze the effect of our differentiable Procrustes loss. In Table 1, we compare several different alignment losses. As a baseline, we train our model to directly regress the 9DoF alignment parameters with an ℓ_2 loss. We then evaluate our approach with (final) and without (no Procrustes) our differentiable Procrustes loss. For CAD model alignment to 3D scans, our differentiable Procrustes alignment notably improves performance, by over 14.98%.

How much does SOC prediction help? We evaluate our SOC prediction on CAD model alignment in Table 1. We train our model with (final) and without (no SOCs) SOC prediction as well as with coordinate correspondence prediction but without symmetry (no symmetry). We observe that our SOC prediction significantly improves performance, by over 20.75%. Establishing SOCs is fundamental to our approach, as dense correspondences can produce more reliable alignment, and unresolved symmetries can lead to ambiguities and inconsistencies in finding ob-

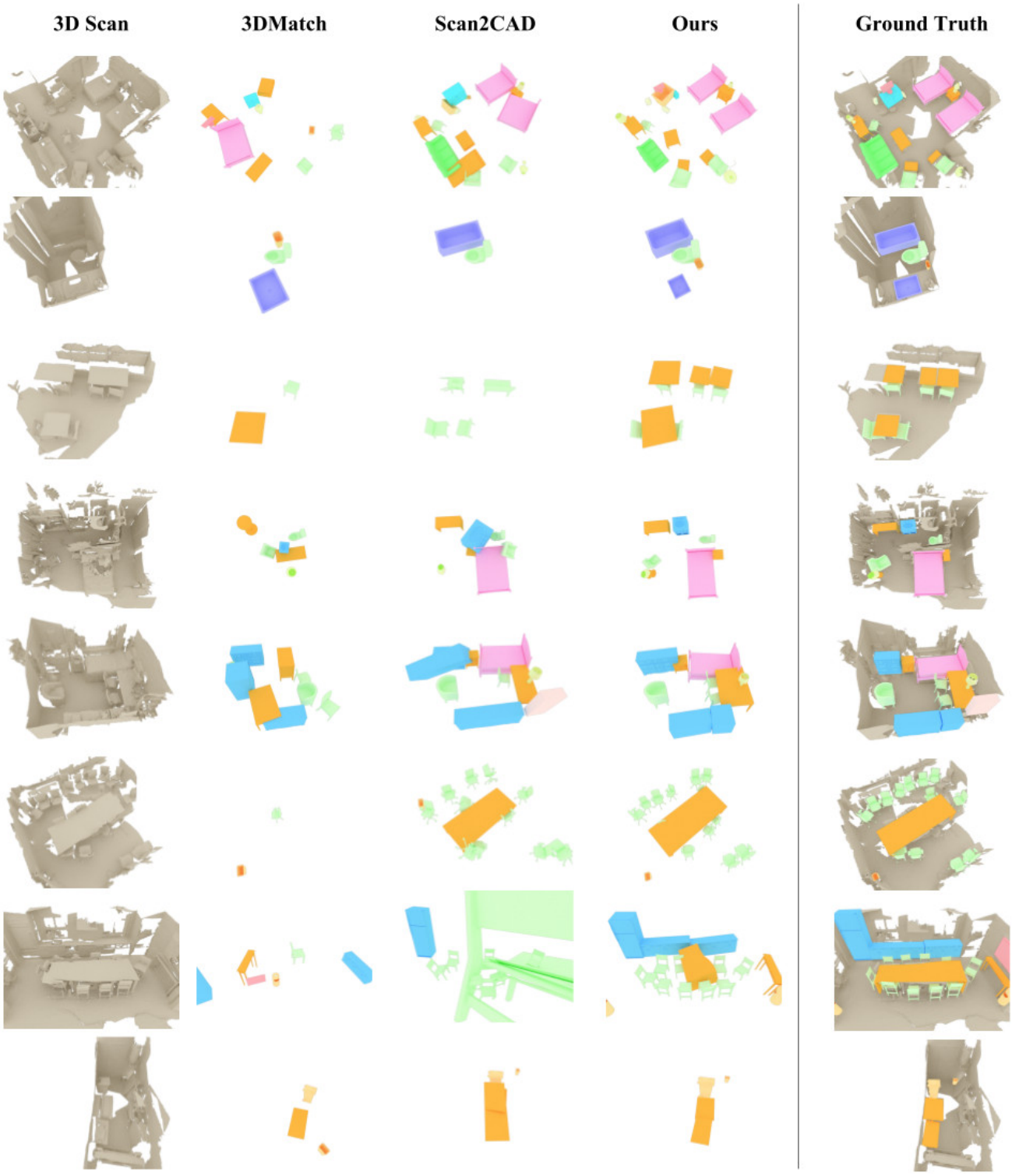

Figure 3: Qualitative comparison of CAD model alignment to ScanNet [7] scans. Our joint formulation of SOC correspondence prediction and differentiable Procrustes alignment enable both more accurate and robust CAD model alignment estimation across varying scene types and sizes.

ject correspondences. In particular, we also evaluate the effect of symmetry classification in our SOCs; explicitly predicting symmetry yields a performance improvement of 10.21%.

What is the effect of using an anchor mechanism for object detection? In Table 1, we also compare our CAD model alignment approach with (final) and without (no anchor) using anchors for object detection, where without anchors we predict only object center locations as a probability heatmap over the volumetric grid of the scan, but do not regress bounding boxes, and thus only crop a fixed neighborhood for the following SOCs and alignment. We observe that by employing bounding box regression, we can improve CAD model alignment performance, as this facilitates scale estimation and allows correspondence features to encompass the full object region.

5.1. Limitations

Although our approach shows significant improvements compared to state of the art, we believe there are directions for improvement. Currently, we focus on the objects in a scan, but do not consider structural components such as walls and floors. We believe, however, that our method could be expanded to detect and match plane segments in the spirit of structural layout detection such as PlaneRCNN [24]. In addition, we currently only consider the geometry of the scan or CAD; however, it is an interesting direction to consider finding matching textures in order to better visually match the appearance of a scan. Finally, we hope to incorporate our alignment algorithm in an online system that can work at interactive rates and give immediate feedback to the scanning operator.

6. Conclusion

We have presented an end-to-end approach that automatically aligns CAD models with commodity 3D scans, which is facilitated with symmetry-aware correspondences and a differentiable Procrustes algorithm. We show that by jointly training the correspondence prediction with direct, end-to-end alignment, our method is able to outperform existing state of the art by over 19.04% in alignment accuracy. In addition, our approach is roughly $250\times$ faster than previous data-driven approaches and thus could be easily incorporated into an online scanning system. Overall, we believe that this is an important step towards obtaining clean and compact representations from 3D scans, and we hope it will open up future research in this direction.

Acknowledgements

We would like to thank Justus Thies and Jürgen Sturm for valuable feedback. This work is supported by Occipital, the ERC Starting Grant Scan2CAD (804724), a Google Faculty Award, an Nvidia Professorship Award, and the ZD.B. We would also like to thank the support of the TUM-IAS, funded by the German Excellence Initiative and the European Union Seventh Framework Programme under grant agreement n° 291763, for the TUM-IAS Rudolf Mößbauer Fellowship.

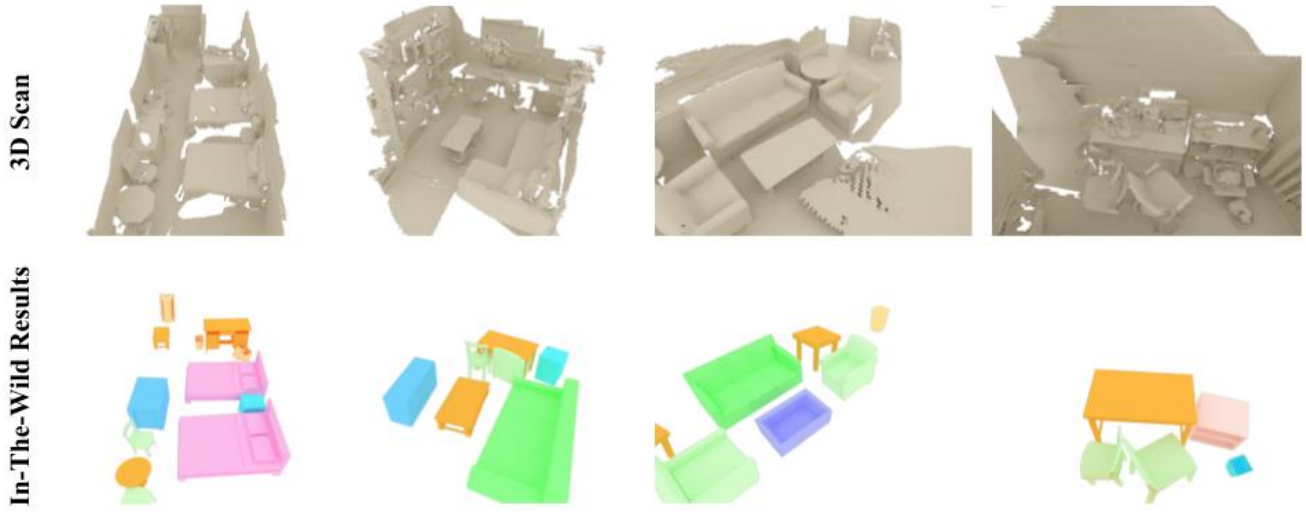

Figure 4: Our end-to-end CAD model alignment approach applied to an unconstrained set of candidate CAD models; here, we use a set of 3000 randomly selected CAD models from ShapeNetCore [4]. The results of our approach (bottom) show robust CAD model alignment performance in a scenario which is often reflected in real-world applications.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. Scan2cad: Learning cad model alignment in rgb-d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 2, 3, 4, 5, 6

[2] Christopher Batty. SDFGen. https://github.com/christopherbatty/SDFGen. 5

[3] Eric Brachmann, Alexander Krull, Sebastian Nowozin, Jamie Shotton, Frank Michel, Stefan Gumhold, and Carsten Rother. Dsac-differentiable ransac for camera localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6684–6692, 2017. 2

[4] 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. 1, 3, 8

[5] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5556–5565. IEEE, 2015. 1, 2

[6] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 2, 5

[7] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 3, 7

[8] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (TOG)*, 36(3):24, 2017. 1, 2

[9] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2018. 5

[10] Haowen Deng, Tolga Birdal, and Slobodan Ilic. Ppf-foldnet: Unsupervised learning of rotation invariant 3d local descriptors. In *European Conference on Computer Vision (ECCV)*. Springer, 2018. 2

[11] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 3d local features for direct pairwise registration. In *Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019. 2

[12] Bertram Drost and Slobodan Ilic. 3d object detection and localization using multimodal point pair features. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*, pages 9–16. IEEE, 2012. 2

[13] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Aligning 3D models to RGB-D images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4731–4740, 2015. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[15] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 5

[16] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 92–101. IEEE, 2016. 3

[17] Binh-Son Hua, Quang-Trung Truong, Minh-Khoi Tran, Quang-Hieu Pham, Asako Kanezaki, Tang Lee, HungYueh Chiang, Winston Hsu, Bo Li, Yijuan Lu, et al. Shrec'17: Rgb-d to cad retrieval with objectnn dataset. 3

[18] Siyuan Huang, Siyuan Qi, Yixin Zhu, Yinxue Xiao, Yuanlu Xu, and Song-Chun Zhu. Holistic 3D scene parsing and reconstruction from a single RGB image. In *European Conference on Computer Vision*, pages 194–211. Springer, 2018. 2

[19] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011. 1, 2

[20] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2422–2431. IEEE, 2017. 2

[21] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *3D Vision-3DV 2013, 2013 International Conference on*, pages 1–8. IEEE, 2013. 2

[22] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3D reconstruction. In *Computer Graphics Forum*, volume 34, pages 435–446. Wiley Online Library, 2015. 2, 6

[23] Joseph J Lim, Hamed Pirsiavash, and Antonio Torralba. Parsing ikea objects: Fine pose estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2992–2999, 2013. 2, 3

[24] Chen Liu, Kihwan Kim, Jinwei Gu, Yasutaka Furukawa, and Jan Kautz. Planercnn: 3d plane detection and reconstruction from a single image. *arXiv preprint arXiv:1812.04072*, 2018. 8

[25] Liangliang Nan, Ke Xie, and Andrei Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):137, 2012. 2

[26] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th*

IEEE international symposium on, pages 127–136. IEEE, 2011. 1, 2

[27] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (TOG)*, 2013. 1, 2

[28] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin Murphy. Towards accurate multi-person pose estimation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4903–4911, 2017. 4

[29] Quang-Hieu Pham, Minh-Khoi Tran, Wenhui Li, Shu Xiang, Heyu Zhou, Weizhi Nie, Anan Liu, Yuting Su, Minh-Triet Tran, Ngoc-Minh Bui, et al. Shrec'18: Rgb-d object-to-cad retrieval. 3

[30] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 3212–3217. Citeseer, 2009. 2, 6

[31] Renato F Salas-Moreno, Richard A Newcombe, Hauke Strasdat, Paul HJ Kelly, and Andrew J Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1352–1359, 2013. 2

[32] Tianjia Shao, Weiwei Xu, Kun Zhou, Jingdong Wang, Dongping Li, and Baining Guo. An interactive approach to semantic modeling of indoor scenes with an RGBD camera. *ACM Transactions on Graphics (TOG)*, 31(6):136, 2012. 2

[33] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 3, 5

[34] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3D: Dataset and methods for single-image 3D shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018. 2, 3

[35] Federico Tombari, Samuele Salti, and Luigi Di Stefano. Unique signatures of histograms for local surface description. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Computer Vision – ECCV 2010*, pages 356–369, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 6

[36] Federico Tombari, Samuele Salti, and Luigi Di Stefano. A combined texture-shape descriptor for enhanced 3d feature matching. In *2011 18th IEEE International Conference on Image Processing*, pages 809–812. IEEE, 2011. 2, 6

[37] 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. *arXiv preprint arXiv:1901.02970*, 2019. 5

[38] Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. Elasticfusion: Dense slam without a pose graph. *Proc. Robotics: Science and Systems, Rome, Italy*, 2015. 1, 2

[39] Yu Xiang, Wonhui Kim, Wei Chen, Jingwei Ji, Christopher Choy, Hao Su, Roozbeh Mottaghi, Leonidas Guibas, and Silvio Savarese. Objectnet3d: A large scale database for 3D object recognition. In *European Conference on Computer Vision*, pages 160–176. Springer, 2016. 3

[40] Yu Xiang, Roozbeh Mottaghi, and Silvio Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 75–82. IEEE, 2014. 3

[41] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 199–208. IEEE, 2017. 2, 6

[42] Chuhang Zou, Ruiqi Guo, Zhizhong Li, and Derek Hoiem. Complete 3D scene parsing from an RGBD image. *International Journal of Computer Vision (IJCV)*, 2018. 2