Scan2CAD: Learning CAD Model Alignment in RGB-D Scans

Armen Avetisyan[1] Manuel Dahnert[1] Angela Dai[1] Manolis Savva[2]
Angel X. Chang[2] Matthias Nießner[1]

[1]Technical University of Munich [2]Simon Fraser University

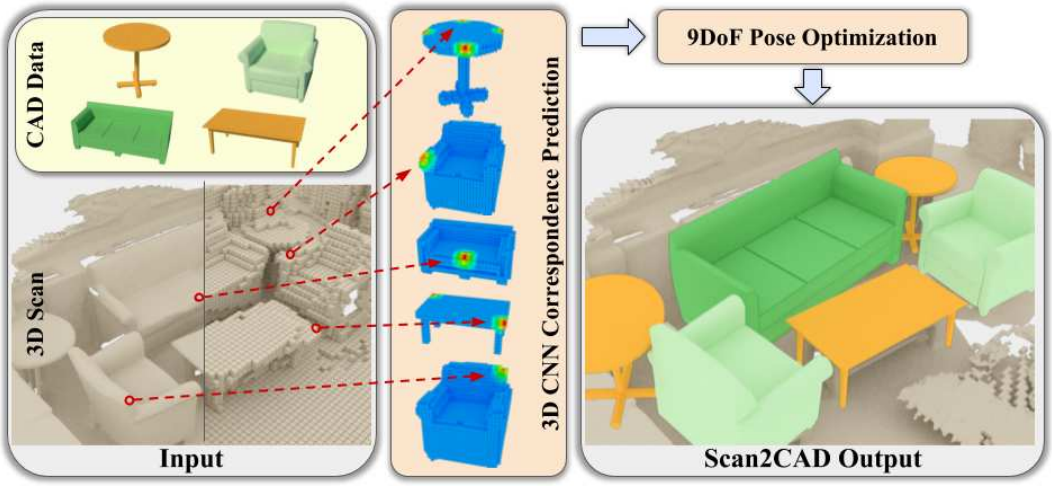


Figure 1: Scan2CAD takes as input an RGB-D scan and a set of 3D CAD models (left). We then propose a novel 3D CNN approach to predict heatmap correspondences between the scan and the CAD models (middle). From these predictions, we formulate an energy minimization to find optimal 9 DoF object poses for CAD model alignment to the scan (right).

Abstract

We present Scan2CAD[1], a novel data-driven method that learns to align clean 3D CAD models from a shape database to the noisy and incomplete geometry of an RGB-D scan. For a 3D reconstruction of an indoor scene, our method takes as input a set of CAD models, and predicts a 9DoF pose that aligns each model to the underlying scan geometry. To tackle this problem, we create a new scan-to-CAD alignment dataset based on 1506 ScanNet scans with 97607 annotated keypoint pairs between 14225 CAD models from ShapeNet and their counterpart objects in the scans. Our method selects a set of representative keypoints in a 3D scan for which we find correspondences to the CAD geometry. To this end, we design a novel 3D CNN architecture to learn a joint embedding between real and synthetic objects, and thus predict a correspondence heatmaps. Based on these correspondence heatmaps, we formulate a variational energy minimization that aligns a given set of CAD models to the reconstruction. We evaluate our approach on our newly introduced Scan2CAD benchmark where we outperform both handcrafted feature descriptor as well as state-of-the-art CNN based methods by 21.39%.

[1]The Scan2CAD dataset is publicly released along with an automated benchmark script for testing under www.Scan2CAD.org

1. Introduction

In recent years, the wide availability of consumer-grade RGB-D sensors, such as the Microsoft Kinect, Intel Real Sense, or Google Tango, has led to significant progress in RGB-D reconstruction. We now have 3D reconstruction frameworks, often based on volumetric fusion [6], that achieve impressive reconstruction quality [18, 29, 30, 40, 21] and reliable global pose alignment [40, 5, 8]. At the same time, deep learning methods for 3D object classification and semantic segmentation have emerged as a primary consumer of large-scale annotated reconstruction datasets [7, 2]. These developments suggest great potential in the future of 3D digitization, for instance, in applications for virtual and augmented reality.

Despite these improvements in reconstruction quality, the geometric completeness and fine-scale detail of indoor scene reconstructions remain a fundamental limitation. In contrast to artist-created computer graphics models, 3D scans are noisy and incomplete, due to sensor noise, motion blur, and scanning patterns. Learning-based approaches for object and scene completion [9, 37, 10] cannot reliably recover sharp edges or planar surfaces, resulting in quality far from artist-modeled 3D content.

One direction to address this problem is to retrieve a set of CAD models from a shape database and align them to an input scan, in contrast to a bottom-up reconstruction of

the scene geometry. If all objects are replaced in this way, we obtain a clean and compact scene representation, precisely serving the requirements for many applications ranging from AR/VR scenarios to architectural design. Unfortunately, matching CAD models to scan geometry is an extremely challenging problem: While high-level geometric structures might be similar, the low-level geometric features differ significantly (e.g., surface normal distributions). This severely limits the applicability of handcrafted geometric features, such as FPFH [33], SHOT [35], point-pair-features [11], or SDF-based feature descriptors [25]. While learning-based approaches like random forests [28, 36] exist, their model capacity remains relatively low, especially in comparison to more modern methods based on deep learning, which can achieve significantly higher accuracy, but remain at their infancy. We believe this is in large part attributed to the lack of appropriate training data.

In this paper, we make the following contributions:

- We introduce the Scan2CAD dataset, a large-scale dataset comprising 97607 pairwise keypoint correspondences and 9DoF alignments between 14225 instances of 3049 unique synthetic models, between ShapeNet [3] and reconstructed scans in ScanNet [7], as well as oriented bounding boxes for each object.
- We propose a novel 3D CNN architecture that learns a joint embedding between real and synthetic 3D objects to predict accurate correspondence heatmaps between the two domains.
- We present a new variational optimization formulation to minimize the distance between scan keypoints and their correspondence heatmaps, thus obtaining robust 9DoF scan-to-CAD alignments.

2. Related work

RGB-D Scanning and Reconstruction The availability of low-cost RGB-D sensors has led to significant research progress in RGB-D 3D reconstruction. A very prominent line of research is based on volumetric fusion [6], where depth data is integrated in a volumetric signed distance function. Many modern real-time reconstruction methods, such as KinectFusion [18, 29], are based on this surface representation. In order to make the representation more memory-efficient, octree [4] or hash-based scene representations have been proposed [30, 21]. An alternative fusion approach is based on points [22]; the reconstruction quality is slightly lower, but it has more flexibility when handling scene dynamics and can be adapted on-the-fly for loop closures [40]. Very recent RGB-D reconstruction frameworks combine efficient scene representations with global pose estimation [5], and can even perform online updates with global loop closures [8]. A closely related direction to ours (and a possible application) is recognition of objects as a part of a SLAM method, and using the retrieved objects as part of a global pose graph optimization [34, 27].

3D Features for Shape Alignment and Retrieval Geometric features have a long-established history in computer vision, such as Spin Images [20], Fast Point Feature Histograms (FPFH) [33], or Point-Pair Features (PPF) [11]. Based on these descriptors or variations of them, researchers have developed shape retrieval and alignment methods. For instance, Kim et al. [24] learn a shape prior in the form of a deformable part model from input scans to find matches at test time; or AA2h [23] use a similar approach to PPF, where a histogram of normal distributions of sample points is used for retrieval. Li et al. [25] propose a formulation based on a hand-crafted TSDF feature descriptor to align CAD models in real-time to RGB-D scans. While these retrieval approaches based on hand-crafted geometric features show initial promise, they struggle to generalize matching between the differing data characteristics of clean CAD models and noisy, incomplete real-world data.

An alternative direction is learned geometric feature descriptors. For example, Nan et al. [28] use a random decision forest to classify objects on over-segmented input geometry from high-quality scans. Shao et al. [36] introduce a semi-automatic system to resolve segmentation ambiguities, where a user first segments a scene into semantic regions, and then shape retrieval is applied. 3DMatch [43] leverage a Siamese neural network to match keypoints in 3D scans for pose estimation. Zhou et al. [44] is of similar nature, proposing a view consistency loss for 3D keypoint prediction network on RGB-D image data. Inspired by such approaches, we develop a 3D CNN-based approach targeting correspondences between the synthetic domain of CAD models and the real domain of RGB-D scan data.

Other approaches retrieve and align CAD models given single RGB [26, 19, 38, 17] or RGB-D [12, 45] images. These methods are related, but our focus is on geometric alignment independent of RGB information, rather than CAD-to-image.

Shape Retrieval Challenges and RGB-D Datasets Shape retrieval challenges have recently been organized as part of the Eurographics 3DOR [16, 32]. Here, the task was formulated as matching of object instances from ScanNet [7] and SceneNN [15] to CAD models from the ShapeNetSem dataset [3]. Evaluation only considered binary in-category vs out-of-category (and sub-category) match as the notion of relevance. As such, this evaluation does not address the alignment quality between scan objects and CAD models, which is our focus.

ScanNet [7] provides aligned CAD models for a small subset of the annotated object instances (for only 200 objects out of the total 36000). Moreover, the alignment

quality is low with many object category mismatches and alignment errors, as the annotation task was performed by crowdsourcing. The PASCAL 3D+ [42] dataset annotates 13898 objects in the PASCAL VOC images with coarse 3D poses defined against representative CAD models. ObjectNet3D [41] provides a dataset of CAD models aligned to 2D images, approximately 200K object instances in 90K images. The IKEA objects [26] and Pix3D [38] datasets similarly provide alignments of a small set of identifiable CAD models to 2D images of the same objects in the real world; the former has 759 images annotated with 90 models, the latter has 10069 annotated with 395 models.

No existing dataset provides fine-grained object instance alignments at the scale of our Scan2CAD dataset with 14225 CAD models (3049 unique instances) annotated to their scan counterpart distributed on 1506 3D scans.

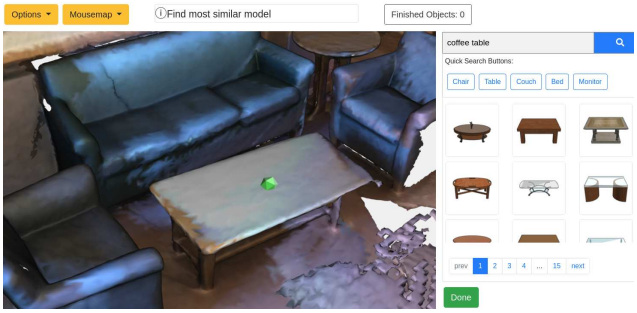
3. Overview

Task We address alignment between clean CAD models and noisy, incomplete 3D scans from RGB-D fusion, as illustrated in Fig. 1. Given a 3D scene $\mathbb{S}$ and a set of 3D CAD models $\mathbb{M} = \{m_i\}$, the goal is to find a 9DoF transformation T_i (3 degrees for translation, rotation, and scale each) for every CAD model m_i such that it aligns with a semantically matching object $\mathbb{O} = \{o_j\}$ in the scan. One important note is that we cannot guarantee the existence of 3D models which exactly matches the geometry of the scan objects.

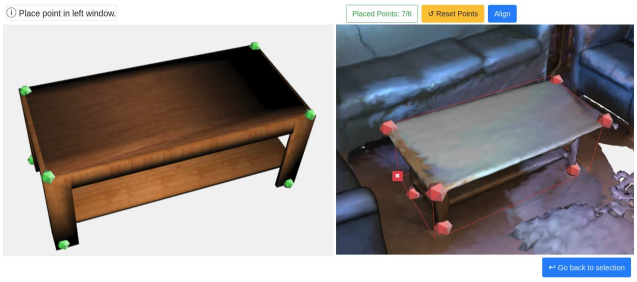
Dataset and Benchmark In Sec. 4, we introduce the construction of our Scan2CAD dataset. We propose an annotation pipeline designed for use by trained annotators. An annotator first inspects a 3D scan and selects a model from a CAD database that is geometrically similar to a target object in the scan. Then, for each model, the annotator defines corresponding keypoint pairs between the model and the object in the scan. From these keypoints, we compute ground truth 9DoF alignments. We annotate the entire ScanNet dataset and use the original training, validation, and test splits to establish our alignment benchmark.

Heatmap Prediction Network In Sec. 5, we propose a 3D CNN taking as input a volume around a candidate keypoint in a scan and a volumetric representation of a CAD model. The network is trained to predict a correspondence heatmap over the CAD volume, representing the likelihood that the input keypoint in the scan is matching with each voxel. The heatmap prediction is formulated as a classification problem, which is easier to train than regression, and produces sparse correspondences needed for pose optimization.

Alignment Optimization Sec. 6 describes our variational alignment optimization. To generate candidate correspondence points in the 3D scan, we detect Harris keypoints, and predict correspondence heatmaps for each Harris keypoint and CAD model. Using the predicted heatmaps we find optimal 9DoF transformations. False alignments are pruned via a geometric confidence metric.



(a) First step: Retrieval view.



(b) Second step: Alignment view.

Figure 2: Our annotation web interface is a two-step process. (a) After the user places an anchor on the scan surface, class-matching CAD models are displayed on the right. (b) Then the user annotates keypoint pairs between the scan and CAD model from which we derive the ground truth 9DoF transformation.

4. Dataset

Our Scan2CAD dataset builds upon the 3D scans from ScanNet [7] and CAD models from ShapeNet [3]. Each *scene* $\mathbb{S}$ contains multiple *objects* $\mathbb{O} = \{o_i\}$, where each *object* o_i is matched with a ShapeNet CAD model m_i and both share multiple keypoint pairs (correspondences) and one transformation matrix T_i defining the alignment. Note that ShapeNet CAD models have a consistently defined front and upright orientation which induces an amodal tight oriented bounding box for each scan object, see Fig. 3.

4.1. Data Annotation

The annotation is done via a web application that allows for simple scaling and distribution of annotation jobs; see Fig. 2. The annotation process is separated into two steps. The first step is object *retrieval*, where the user clicks on a point on the 3D scan surface, implicitly determining an object category label from the ScanNet object instance annotations. We use the instance category label as query text in the ShapeNet database to retrieve and display all matching

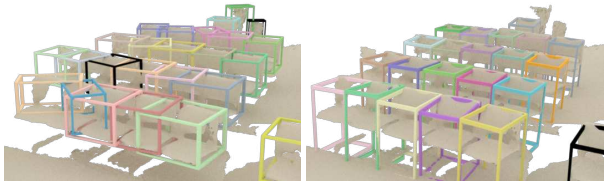


Figure 3: (Left) Oriented bounding boxes (OBBs) computed from the instance segmentation of ScanNet [7] are often incomplete due to missing geometry (e.g., in this case, missing chair legs). (Right) Our OBBs are derived from the aligned CAD models and are thus complete.

CAD models in a separate window as illustrated in Fig. 2a. After selecting a CAD model the user performs *alignment*.

In the alignment step, the user sees two separate windows in which the CAD model (left) and the scan object (right) are shown (see Fig. 2b). Keypoint correspondences are defined by alternately clicking paired points on the CAD model and scan object. We require users to specify at least 6 keypoint pairs to determine a robust ground truth transformation. After keypoint pairs are specified, the alignment computation is triggered by clicking a button. This alignment (given exact 1-to-1 correspondences) is solved with the genetic algorithm *CMA-ES* [14, 13] that minimizes the point-to-point distance over 9 parameters. In comparison to gradient-based methods or Procrustes superimposition method, we found this approach to perform significantly better in reliably returning high-quality alignments regardless of initialization.

The quality of these keypoint pairs and alignments was verified in several verification passes, with re-annotations performed to ensure a high quality of the dataset. The verification passes were conducted by the authors of this work.

A subset of the ShapeNet CAD models have symmetries that play an important role in making correspondences. Hence, we annotated all ShapeNet CAD models used in our dataset with their rotational symmetries to prevent false negatives in evaluations. We defined 2-fold (C_2), 4-fold (C_4) and infinite (C_∞) rotational symmetries around a canonical axis of the object.

4.2. Dataset Statistics

The annotation process yielded 97607 keypoint pairs on 14225 (3049 unique) CAD models with their respective scan counterpart distributed on a total of 1506. Approximately 28% out of the 3049 CAD models have a symmetry tag (either C_2, C_4 or C_∞).

Given the complexity of the task and to ensure high quality annotations, we employed 7 part-time annotators (in contrast to crowd-sourcing). On average, each scene has been edited 1.76 times throughout the re-annotation cycles. The top 3 annotated model classes are chairs, tables and cabinets which arises due to the nature of indoor scenes in ScanNet. The number of objects aligned per scene ranges from 1 to 40 with an average of 9.3. It took annotators on average of 2.48min to align each object, where the time to find an appropriate CAD model dominated the time for keypoint placement. The average annotation time for an entire scene is 20.52min.

It is interesting to note that manually placed keypoint correspondences between scans and CAD models differ significantly from those extracted from a Harris corner detector. Here, we compare the mean distance from the annotated CAD keypoint to: (1) the corresponding annotated scan keypoint ($= 3.5cm$) and (2) the nearest Harris keypoint in the scan ($= 12.8cm$).

4.3. Benchmark

Using our annotated dataset, we designed a benchmark to evaluate scan-to-CAD alignment methods. A model alignment is considered successful only if the category of the CAD model matches that of the scan object *and* the pose error is within translation, rotational, and scale bounds relative to the ground truth CAD. We do not enforce strict instance matching (i.e., matching the exact CAD model of the ground truth annotation) as ShapeNet models typically do not identically match real-world scanned objects. Instead, we treat CAD models of the same category as interchangeable (according to the ShapeNetCorev2 *top-level synset*).

Once a CAD model is determined to be aligned correctly, the ground truth counterpart is removed from the candidate pool in order to prevent multiple alignments to the same object. Alignments are fully parameterized by 9 pose parameters. A quantitative measure based on bounding box overlap (IoU) can be readily calculated with these parameters as CAD models are defined on the unit box. The error thresholds for a successful alignment are set to $\epsilon_t \leq 20$cm, $\epsilon_r \leq 20°$, and $\epsilon_s \leq 20\%$ for translation, rotation, and scale respectively (for extensive error analysis please see the supplemental). The rotation error calculation takes C_2, C_4 and C_∞ rotated versions into account.

The Scan2CAD dataset and associated symmetry annotations are available to the community. For standardized comparison of future approaches, we operate an automated test script on a hidden test set that can be found under www.Scan2CAD.org.

5. Correspondence Prediction Network

5.1. Data Representation

Scan data is represented by its signed distance field (SDF) encoded in a volumetric grid and generated through *volumetric fusion* [6] from the depth maps of the RGB-D reconstruction (voxel resolution = $3cm$, truncation = $15cm$). For the CAD models, we compute unsigned distance fields (DF) using the level-set generation toolkit by Batty [1].

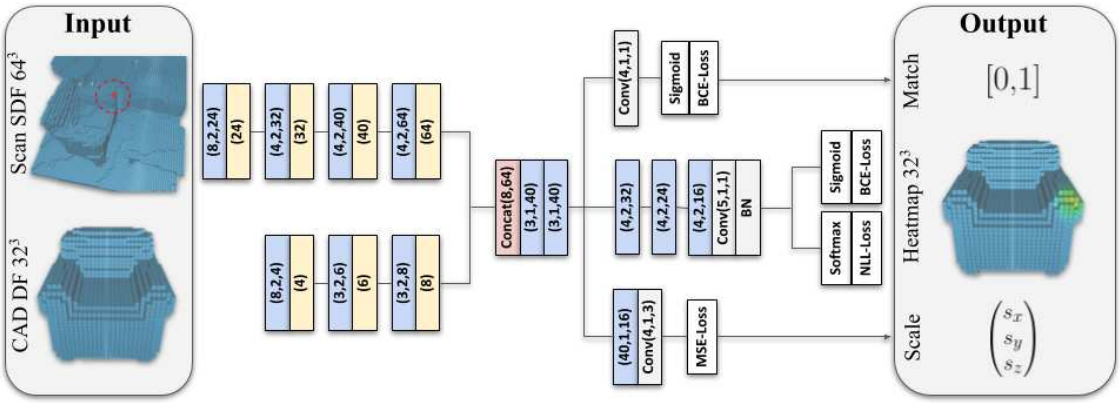


Figure 4: 3D CNN architecture of our Scan2CAD approach: we take as input SDF chunks around a given keypoint from a 3D scan and the DF of a CAD model. These are encoded with 3D CNNs to learn a shared embedding between the synthetic and real data; from this, we classify whether there is semantic compatibility between both inputs (top), predict a correspondence heatmap in the CAD space (middle) and the scale difference between the inputs (bottom).

5.2. Network Architecture

Our architecture takes as input a pair of voxel grids: A SDF centered at a point in the scan with a large receptive field at 64^3 size, and a DF of a particular CAD model at 32^3 size. We use a series of convolutional layers to separately encode each input stream (see Fig. 4). The two encoders compress the volumetric representation into compact feature volumes of $4^3 \times 64$ (scan) and $4^3 \times 8$ (CAD) which are then concatenated before passing to the decoder stage. The decoder stage predicts three output targets, heatmap, compatibility, and scale, described as follows:

Heatmap The first output is a heatmap $H : \Omega \rightarrow [0, 1]$ over the 32^3 voxel domain $\Omega \in \mathbb{N}^3$ of the CAD model producing the voxel-wise correspondence probability. This indicates the probability of matching each voxel in Ω to the center point of the scan SDF. We train our network using a combined binary cross-entropy (BCE) loss and a negative log-likelihood (NLL) to predict the final heatmap H. The raw output $S : \Omega \rightarrow \mathbb{R}$ of the last layer in the decoder is used to generate the heatmaps:

$$H_1 : \Omega \rightarrow [0, 1], \quad x \mapsto \text{sigmoid}(S(x))$$
$$H_2 : \Omega \rightarrow [0, 1], \quad x \mapsto \text{softmax}(S(x))$$
$$\mathcal{L}_H = \sum_{x \in \Omega} w(x) \cdot \text{BCE}(H_1, H_{\text{GT}}) + \sum_{x \in \Omega} v \cdot \text{NLL}(H_2, H_{\text{GT}})$$

where $w(x) = 64.0$ if $H_{\text{GT}}(x) > 0.0$ else $1.0, v = 64$ are weighting factors to increase the signal of the few sparse positive keypoint voxels in the voxel grid ($\approx 99\%$ of the target voxels have a value equal to 0). The combination of the sigmoid and softmax terms is a compromise between high recall but low precision using sigmoid, and more locally sharp keypoint predictions using softmax over all voxels. The final target heatmap, used later for alignment, is constructed with an element-wise multiplication of both heatmap variations: $H = H_1 \circ H_2$.

Compatibility The second prediction target is a single probability score $\in [0, 1]$ indicating semantic compatibility between scan and CAD. This category equivalence score is 0 when the category labels are different (e.g., scan table and CAD chair) and 1 when the category labels match (e.g., scan chair and CAD chair). The loss function for this output is a sigmoid function followed by a BCE loss:

$$\mathcal{L}_{\text{compat.}} = \text{BCE}(\text{sigmoid}(x), x_{\text{GT}})$$

Scale The third output predicts the scale $\in \mathbb{R}^3$ of the CAD model to the respective scan. Note that we do not explicitly enforce positivity of the predictions. This loss term is a mean-squared-error (MSE) for a prediction $x \in \mathbb{R}^3$:

$$\mathcal{L}_{\text{scale}} = \text{MSE}(x, x_{\text{GT}}) = \|x - x_{\text{GT}}\|_2^2$$

Finally, to train our network, we use a weighted combination of the presented losses:

$$\mathcal{L} = 1.0\mathcal{L}_{\text{H}} + 0.1\mathcal{L}_{\text{compat.}} + 0.2\mathcal{L}_{\text{scale}}$$

where the weighting of each loss component was empirically determined for balanced convergence.

5.3. Training Data Generation

Voxel Grids Centered scan volumes are generated by projecting the annotated keypoint into the scan voxel grid and then cropping around it with a crop window of 63^3. Ground truth heatmaps are generated by projecting annotated keypoints (and any symmetry-equivalent keypoints) into the CAD voxel grid. We then use a Gaussian blurring kernel ($\sigma = 2.0$) on the voxel grid to account for small keypoint annotation errors and to avoid sparsity in the loss residuals.

Training Samples With our annotated dataset we generate $N_{P,\text{ann.}} = 97607$ positive training pairs where one pair consists of an annotated scan keypoint and the corresponding CAD model. Additionally, we create $N_{P,\text{aug.}} = 10 \cdot N_{P,\text{ann.}}$, augmented positive keypoint pairs by randomly sampling points on the CAD surface, projecting them to the scan via the ground truth transformation and rejecting if the distance to the surface in the scan $\geq 3cm$. In total we generate $N_P = N_{P,\text{ann.}} + N_{P,\text{aug.}}$ positive training pairs.

Negative pairs are generated in two ways: (1) Randomly choosing a voxel point in the scan and a random CAD model (likelihood of false negative is exceedingly low). (2) Taking an annotated scan keypoint and pairing it with a random CAD model of different class. We generate $N_N = N_P$ negative samples with (1) and $N_{HN} = N_P$ with (2).

Hence, the training set has a positives-to-negatives ratio of 1:2 ($N_P : N_N + N_{HN}$). We found an over-representation of negative pairs gives satisfactory performance on the compatibility prediction.

5.4. Training Process

We use an SGD optimizer with a batch size of 32 and an initial learning rate of 0.01, which is decreased by $1/2$ every 50K iterations. We train for 250K iterations (≈ 62.5 hours). The weights are initialized randomly. The losses of the heatmap prediction stream and the scale prediction stream are masked such that only positive samples make up the residuals for back-propagation.

The CAD encoder is pre-trained with an auto-encoder on ShapeNet models with a reconstruction task and a MSE as loss function. All models of ShapeNetCore ($\approx 55K$) are used for pre-training and the input and output dimensions are 32^3 distance field grids. The network is trained with SGD until convergence (≈ 50 epochs).

6. Alignment Optimization

Filtering The input to our alignment optimization is a representative set of Harris keypoints $\mathbb{K} = \{p_j\}$, $j = 1 \dots N_0$ from a scene $\mathbb{S}$ and a set of CAD models $\mathbb{M} = \{m_i\}$. The correspondences between $\mathbb{K}$ and $\mathbb{M}$ were established by the correspondence prediction from the previous stage (see Sec. 5) where each keypoint p_j is tested against every model m_i.

Since not every keypoint p_j semantically matches to every CAD model m_i, we reject correspondences based on the compatibility prediction of our network. The threshold for rejecting p_j is determined by the Otsu thresholding scheme [31]. In practice this method turned out to be much more effective than a fixed threshold. After the filtering there are $N \leq N_0$ (usually $N \approx 0.1N_0$) correspondence pairs to be used for the alignment optimization.

Variational Optimization From the remaining $\mathbb{K}_{\text{filter.}} \subset \mathbb{K}$ Harris keypoints, we construct *point-heatmap* pairs (p_j, H_j) for each CAD model m_i, with $p_j \in \mathbb{R}^3$ a point in the scan and $H_j : \Omega \to [0, 1]$ a heatmap.

In order to find an optimal pose we construct the following minimization problem:

$$\mathrm{c}_{\text{vox}} = T_{\text{world}\to\text{vox}} \cdot T_{m_i}(a, s) \cdot p_j$$
$$f = \min_{a,s} \sum_{j}^{N} (1 - H_j(\mathrm{c}_{\text{vox}}))^2 + \lambda_s \|s\|_2^2 \qquad (1)$$

where c_{vox} is a voxel coordinate, $T_{\text{world}\to\text{vox}}$ denotes a transformation that maps world points into the voxel grid for look-ups, a denotes the coordinates of the Lie algebra (for rotation and translation), s defines the scale, and λ_s defines the scale regularization strength. a, s compose a transformation matrix $T_{m_i} = \psi(a_{m_i}, s_{m_i})$:

$$\psi : \mathbb{R}^6 \times \mathbb{R}^3 \to \mathbb{R}^{4\times4},$$
$$a, s \mapsto \text{expm}\left(\begin{bmatrix} \Gamma(a_{1,2,3}) & a_{4,5,6} \\ 0 & 0 \end{bmatrix}\right) \cdot \begin{bmatrix} s & 0 \\ 0 & 1 \end{bmatrix}$$

where Γ is the hat map, expm is the matrix exponential.

We solve Eq. 1 using the Levenberg-Marquardt (LM) algorithm. As we can suffer from zero-gradients (especially at bad initialization), we construct a scale-pyramid from the heatmaps which we solve in coarse-to-fine fashion.

In each LM step we optimize over the incremental change and update the parameters as following: $T_{m_i}^{k+1} \leftarrow \phi(a^*, s^*) \cdot T_{m_i}^{k}$ where a^*, s^* are the optimal parameters. As seen in Eq. 1, we add a regularization on the scale in order to prevent degenerate solutions which can appear for very large scales.

By restarting the optimization with different translation parameters (i.e., varying initializations), we obtain multiple alignments per CAD model m_i. We then generate as many CAD model alignments as required for a given scene in the evaluation. Note, in a ground truth scene one unique CAD model m_i can appear in multiple locations e.g., chairs in conference rooms.

Pruning Finally, there will be alignments of various CAD models into a scene where a subset will be misaligned. In order to select only the best alignments and prune potential misalignments we use a confidence metric similar to [25]; for more detail, we refer to the appendix.

7. Results

7.1. Correspondence Prediction

To quantify the performance of correspondence heatmap predictions, we evaluate the voxel-wise F1-score for a prediction and its Gaussian-blurred target. The task is challenging and by design $\frac{2}{3}$ test samples are false correspondences, $\approx 99\%$ of the target voxels are 0-valued, and only a

base [+variations, ...]	bath	bookshelf	cabinet	chair	display	sofa	table	trash bin	other	class avg.	avg.
+sym	46.88	44.39	40.49	64.46	26.85	56.26	47.15	38.43	24.68	43.29	48.01
+sym,+scale	51.35	45.46	45.24	66.94	29.88	64.78	48.30	38.00	28.65	46.51	50.85
+sym,+CP	59.32	51.93	55.11	70.99	41.58	66.77	53.74	43.39	42.93	53.97	60.44
+scale,+CP	45.24	45.85	47.16	61.55	27.65	51.92	41.21	31.13	29.62	42.37	47.64
+sym,+scale,+CP	56.05	51.28	**57.45**	72.64	36.36	70.63	52.28	46.80	43.32	54.09	60.43
+sym,+scale,+CP,+PT (3/3 fix)	57.03	50.63	56.76	70.39	39.74	65.00	52.03	**46.87**	41.83	53.36	58.61
+sym,+scale,+CP,+PT (1/3 fix)	**60.08**	**58.62**	56.35	**73.92**	**44.19**	**75.08**	**56.80**	45.78	**46.53**	**57.48**	**63.94**

Table 1: Correspondence prediction F1-scores in % for variations of our correspondence prediction network. We evaluate the effect of symmetry (sym), predicting scale (scale), predicting compatibility (CP), encoder pre-training (PT), and pre-training with parts of the encoder fixed (#fix), see Sec. 5 for more detail regarding our network design and training scheme.

single 1-valued voxel out of 32^3 voxels exists. The F1-score will increase only by identifying true correspondences. As seen in Tab. 1, our best 3D CNN achieves 63.94%.

Tab. 1 additionally addressed our design choices; in particular, we evaluate the effect of using pre-training (PT), using compatibility (CP) as a proxy loss (defined in Sec. 5.2), enabling symmetry awareness (sym), and predicting scale (scale). Here, a pre-trained network reduces overfitting, enhancing generalization capability. Optimizing for compatibility strongly improves heatmap prediction as it efficiently detects false correspondences. While predicting scale only slightly influences the heatmap predictions, it becomes very effective for the later alignment stage. Additionally, incorporating symmetry enables significant improvement by explicitly disambiguating symmetric keypoint matches.

7.2. Alignment

In the following, we compare our approach to other handcrafted feature descriptors: FPFH [33], SHOT [39], Li et al. [25] and a learned feature descriptor: 3DMatch [43] (trained on our Scan2CAD dataset). We combine these descriptors with a RANSAC outlier rejection method to obtain pose estimations for an input set of CAD models. A detailed description of the baselines can be found in the appendix. As seen in Tab. 2, our best method achieves 31.68% and outperforms all other methods by a significant margin. We additionally show qualitative results in Fig. 5. Compared to state-of-the-art handcrafted feature descriptors, our learned approach powered by our Scan2CAD dataset produces considerably more reliable correspondences and CAD model alignments. Even compared to the learned descriptor approach of 3DMatch, our explicit learning across the synthetic and real domains coupled with our alignment optimization produces notably improved CAD model alignment.

Fig. 6 shows the capability of our method to align in an unconstrained real-world setting where ground truth CAD models are not given, we instead provide a set of 400 random CAD models from ShapeNet [3].

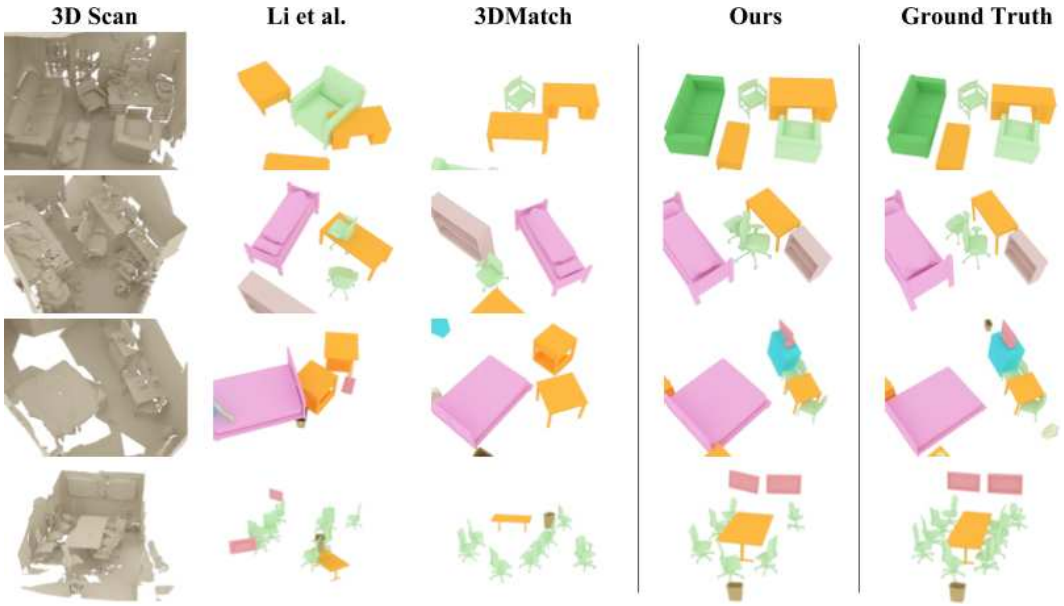


Figure 5: Qualitative comparison of alignments on four different test ScanNet [7] scenes. Our approach to learning geometric features between real and synthetic data produce much more reliable keypoint correspondences, which coupled with our alignment optimization, produces significantly more accurate alignments.

	bath	bookshelf	cabinet	chair	display	sofa	table	trash bin	other	class avg.	avg.
FPFH (Rusu et al. [33])	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. [39])	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. [25]	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. [43])	0.00	5.67	2.86	21.25	2.41	10.91	6.98	3.62	4.65	6.48	10.29
Ours: +sym	24.30	10.61	5.97	9.49	3.90	25.26	12.34	10.74	3.58	11.80	8.772
Ours: +sym,+scale	18.99	13.61	7.24	14.73	9.76	41.05	14.04	5.26	6.29	14.55	11.48
Ours: +sym,+CP	35.90	32.35	28.64	40.48	18.85	60.00	33.11	28.42	16.89	32.74	29.42
Ours: +scale,+CP	34.18	31.76	21.82	37.02	14.75	50.53	32.31	**31.05**	11.59	29.45	26.75
Ours: +sym,+scale,+CP	36.20	**36.40**	**34.00**	**44.26**	17.89	**70.63**	30.66	30.11	20.60	**35.64**	**31.68**
Ours: +sym,+scale,+CP,+PT (3/3 fix)	**37.97**	30.15	28.64	41.55	19.51	57.89	33.85	20.00	17.22	31.86	29.27
Ours: +sym,+scale,+CP,+PT (1/3 fix)	34.81	**36.40**	29.00	40.60	**23.25**	66.00	**37.64**	24.32	**22.81**	34.98	31.22

Table 2: Accuracy comparison (%) on our CAD alignment benchmark. While handcrafted feature descriptors can achieve some alignment on more featureful objects (e.g., chairs, sofas), they do not tolerate well the geometric discrepancies between scan and CAD data – which remains difficult for the learned keypoint descriptors of 3DMatch. Scan2CAD directly addresses this problem of learning features that generalize across these domains, thus significantly outperforming state of the art.

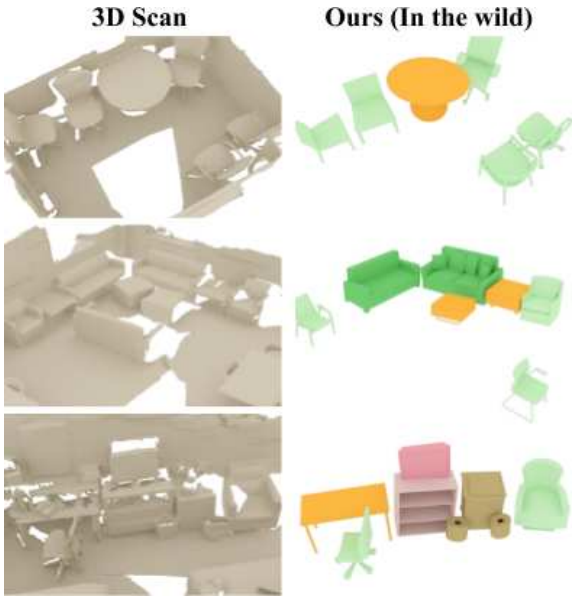


Figure 6: Unconstrained scenario where instead of having a ground truth set of CAD models given, we use a set of 400 randomly selected CAD models from ShapeNetCore [3], more closely mimicking a real-world application scenario.

8. Limitations

While the focus of this work is mainly on the alignment between 3D scans and CAD models, we only provide a basic algorithmic component for retrieval (finding the most similar model). This necessitates an exhaustive search over a set of CAD models. We believe that one of the immediate next steps in this regard would be designing a neural network architecture that is specifically trained on shape similarity between scan and CAD geometry to introduce more efficient CAD model retrieval. Additionally, we currently only consider geometric information, and it would also be intresting to introduce learned color features into the correspondence prediction, as RGB data is typically higher-resolution than depth or geometry, and could potentially improve alignment results.

9. Conclusion

In this work, we presented Scan2CAD, which aligns a set of CAD models to 3D scans by predicting correspondences in form of heatmaps and then optimizes over these correspondence predictions. First, we introduce a new dataset of 9DoF CAD-to-scan alignments with 97607 pairwise keypoint annotations defining the alignment of 14225 objects. Based on this new dataset, we design a 3D CNN to predict correspondence heatmaps between a CAD model and a 3D scan. From these predicted heatmaps, we formulate a variational cost minimization that then finds the optimal 9DoF pose alignments between CAD models and the scan, enabling effective transformation of noisy, incomplete RGB-D scans into a clean, complete CAD model representation. This enables us to achieve significantly more accurate results than state-of-the-art approaches, and we hope that our dataset and benchmark will inspire future work towards bringing RGB-D scans to CAD or artist-modeled quality.

Acknowledgements

We would like to thank the expert annotators Soh Yee Lee, Rinu Shaji Mariam, Suzana Spasova, Emre Taha, Sebastian Thekkekara, and Weile Weng for their efforts in building the Scan2CAD dataset. We thank valuable discussions with Jürgen Sturm. This work is supported by Occipital, the ERC Starting Grant Scan2CAD (804724), a Google Faculty 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 and Hans-Fisher Fellowship (Focus Group Visual Computing).

References

[1] C. Batty. SDFGen. https://github.com/christopherbatty/SDFGen. 4

[2] A. Chang, A. Dai, T. Funkhouser, M. Halber, M. Niessner, M. Savva, S. Song, A. Zeng, and Y. Zhang. Matterport3D: Learning from RGB-D data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1

[3] 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. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 2, 3, 7, 8

[4] J. Chen, D. Bautembach, and S. Izadi. Scalable real-time volumetric surface reconstruction. *ACM Transactions on Graphics (TOG)*, 32(4):113, 2013. 2

[5] S. Choi, Q.-Y. Zhou, and V. 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] B. Curless and M. 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. 1, 2, 4

[7] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 1, 2, 3, 4, 7

[8] 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 Transactions on Graphics (TOG)*, 36(3):24, 2017. 1, 2

[9] A. Dai, C. R. Qi, and M. Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 1

[10] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. *arXiv preprint arXiv:1712.10215*, 2018. 1

[11] B. Drost and S. 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

[12] S. Gupta, P. Arbeláez, R. Girshick, and J. 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

[13] N. Hansen. Benchmarking a bi-population cma-es on the bbob-2009 function testbed. In *Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers*, pages 2389–2396. ACM, 2009. 4

[14] N. Hansen, S. D. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (cma-es). *Evolutionary computation*, 11(1):1–18, 2003. 4

[15] B.-S. Hua, Q.-H. Pham, D. T. Nguyen, M.-K. Tran, L.-F. Yu, and S.-K. Yeung. Scenenn: A scene meshes dataset with annotations. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 92–101. IEEE, 2016. 2

[16] B.-S. Hua, Q.-T. Truong, M.-K. Tran, Q.-H. Pham, A. Kanezaki, T. Lee, H. Chiang, W. Hsu, B. Li, Y. Lu, et al. Shrec17: Rgb-d to cad retrieval with objectnn dataset. 2

[17] S. Huang, S. Qi, Y. Zhu, Y. Xiao, Y. Xu, and S.-C. Zhu. Holistic 3D scene parsing and reconstruction from a single RGB image. In *European Conference on Computer Vision*, pages 194–211. Springer, 2018. 2

[18] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. 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

[19] H. Izadinia, Q. Shan, and S. M. Seitz. Im2cad. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2422–2431. IEEE, 2017. 2

[20] A. E. Johnson. Spin-images: a representation for 3-d surface matching. 1997. 2

[21] O. Kähler, V. A. Prisacariu, C. Y. Ren, X. Sun, P. Torr, and D. Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE transactions on visualization and computer graphics*, 21(11):1241–1250, 2015. 1, 2

[22] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. 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

[23] Y. M. Kim, N. J. Mitra, Q. Huang, and L. Guibas. Guided real-time scanning of indoor objects. In *Computer Graphics Forum*, volume 32, pages 177–186. Wiley Online Library, 2013. 2

[24] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas. Acquiring 3D indoor environments with variability and repetition. *ACM Transactions on Graphics (TOG)*, 31(6):138, 2012. 2

[25] Y. Li, A. Dai, L. Guibas, and M. 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, 7, 8

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

[27] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric object-level slam. In *2018 International Conference on 3D Vision (3DV)*, pages 32–41. IEEE, 2018. 2

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

[29] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and

A. 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

[30] 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

[31] N. Otsu. A threshold selection method from gray-level histograms. *IEEE transactions on systems, man, and cybernetics*, 9(1):62–66, 1979. 6

[32] Q.-H. Pham, M.-K. Tran, W. Li, S. Xiang, H. Zhou, W. Nie, A. Liu, Y. Su, M.-T. Tran, N.-M. Bui, et al. Shrec18: Rgb-d object-to-cad retrieval. 2

[33] R. B. Rusu, N. Blodow, and M. 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, 7, 8

[34] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. 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

[35] S. Salti, F. Tombari, and L. Di Stefano. Shot: Unique signatures of histograms for surface and texture description. *Computer Vision and Image Understanding*, 125:251–264, 2014. 2

[36] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo. An interactive approach to semantic modeling of indoor scenes with an RGBD camera. *ACM Transactions on Graphics (TOG)*, 31(6):136, 2012. 2

[37] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1

[38] X. Sun, J. Wu, X. Zhang, Z. Zhang, C. Zhang, T. Xue, J. B. Tenenbaum, and W. 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

[39] F. Tombari, S. Salti, and L. Di Stefano. Unique signatures of histograms for local surface description. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, pages 356–369, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. 7, 8

[40] T. Whelan, S. Leutenegger, R. F. Salas-Moreno, B. Glocker, and A. J. Davison. Elasticfusion: Dense slam without a pose graph. *Proc. Robotics: Science and Systems, Rome, Italy*, 2015. 1, 2

[41] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese. Objectnet3d: A large scale database for 3D object recognition. In *European Conference on Computer Vision*, pages 160–176. Springer, 2016. 3

[42] Y. Xiang, R. Mottaghi, and S. 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

[43] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. 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, 7, 8

[44] X. Zhou, A. Karpur, C. Gan, L. Luo, and Q. Huang. Unsupervised domain adaptation for 3d keypoint estimation via view consistency. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 137–153, 2018. 2

[45] C. Zou, R. Guo, Z. Li, and D. Hoiem. Complete 3D scene parsing from an RGBD image. *International Journal of Computer Vision (IJCV)*, 2018. 2