FastCAD: Real-Time CAD Retrieval and Alignment from Scans and Videos

Florian Langer[1,2⋆], Jihong Ju[1], Georgi Dikov[1], Gerhard Reitmayr[1], Mohsen Ghafoorian[1]

[1] XR Labs, Qualcomm Technologies, Inc.
[2] Department of Engineering, University of Cambridge
fml35@cam.ac.uk
{jihoju,gdikov,gerhardr,mghafoor}@qti.qualcomm.com

Abstract. Digitising the 3D world into a clean, CAD model-based representation has important applications for augmented reality and robotics. Current state-of-the-art methods are computationally intensive as they individually encode each detected object and optimise CAD alignments in a second stage. In this work, we propose FastCAD, a real-time method that simultaneously retrieves and aligns CAD models for all objects in a given scene. In contrast to previous works, we directly predict alignment parameters and shape embeddings. We achieve high-quality shape retrievals by learning CAD embeddings in a contrastive learning framework and distilling those into FastCAD. Our single-stage method accelerates the inference time by a factor of 50 compared to other methods operating on RGB-D scans while outperforming them on the challenging Scan2CAD alignment benchmark. Further, our approach collaborates seamlessly with online 3D reconstruction techniques. This enables the real-time generation of precise CAD model-based reconstructions from videos at 10 FPS. Doing so, we significantly improve the Scan2CAD alignment accuracy in the video setting from 43.0% to 48.2% and the reconstruction accuracy from 22.9% to 29.6%.

Keywords: online, precise CAD model-based reconstruction

1 Introduction

Representing environments and rooms by aligned 3D CAD models is crucial for many downstream tasks in augmented reality or robotics. Compared to noisy 3D scene meshes or point clouds, a CAD-based representation has many advantages, such as the absence of holes in objects, clean surface geometry, object-level annotations, and potential part-level scene understanding. Additionally, the representation is more compact, with significantly fewer vertices and faces, which allows for faster rendering and collision simulations.

In this work, we introduce FastCAD, which is carefully designed to perform real-time CAD retrieval and alignment (see Fig. 1). First, to achieve this goal, FastCAD simultaneously solves object alignment and retrieval thanks to the proposed embedding distillation technique. For this, we first learn an embedding space by training a separate encoder network in a contrastive learning setting. Noisy, partial scans and clean CAD models are embedded into a unified embedding space. By introducing two auxiliary tasks, performing foreground-background segmentation of the noisy object scan and predicting the similarity of the positive and negative CAD model used for the contrastive setup, we further improve the quality of the learned embeddings. Rather than using the encoder network to

⋆ This work was done as part of an internship at Qualcomm Technologies, Inc.

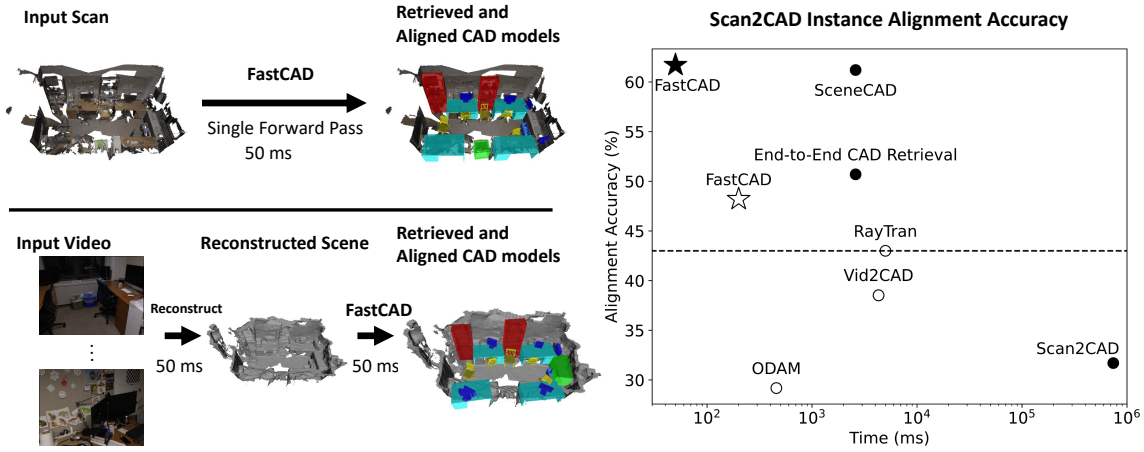

Fig. 1. Intro: FastCAD retrieves and aligns CAD models to point clouds in real-time. FastCAD can either directly operate on an RGB-D scan (left top) or on the output of an off-the-shelf reconstruction method, which takes an RGB video as its input (left bottom). The graph on the right shows the Scan2CAD [3] instance alignment accuracy as a function of inference time compared to competing methods. Note that the inference time is displayed on a log scale. Closed circles and stars (ours) denote methods operating on RGB-D scans, while open circles and stars represent methods using RGB videos as inputs. FastCAD outperforms previous methods in both settings while being significantly faster than the previously fastest methods. Note that RayTran [35] did not disclose their run-times but is most likely much slower than FastCAD (see Supp. Mat.).

obtain embedding vectors at inference time, we distil the embeddings into FastCAD by supervising its shape embedding prediction per detection by the embedding of the ground-truth CAD model. Doing so greatly improves the speed as well as the quality of the retrieved shapes.

Second, FastCAD directly predicts alignment parameters. We parameterise the alignments with oriented 3D bounding boxes where we additionally predict the front-facing side of the CAD model within the bounding box. This is significantly faster than analysis-by-synthesis-based methods [1,16] where CAD alignments are obtained iteratively by minimising rendering-based alignment objectives. It is also more efficient than correspondence-based methods [3,2,4] where the network outputs object-to-CAD correspondences and object poses are extracted with an additional alignment optimisation [8]. At inference time, the shape embeddings predicted by FastCAD are used to retrieve the nearest neighbor CAD models from the embedding space. Those CAD models are aligned inside the predicted bounding boxes according to the predicted front-facing side to form the final output. In this way, we achieve a very efficient method running in just 50 ms per RGB-D scan (compared to [2,4], which takes 2.6 s) while achieving a similar accuracy on the Scan2CAD alignment benchmark compared to [4] (61.7% vs 61.2%, see Fig. 1).

Third, we can use FastCAD in conjunction with reconstruction methods (e.g. [33,18,5]) to perform precise, real-time CAD alignments from videos. For this, we sample a point cloud from the output mesh generated with [18] and use it as the input to FastCAD to predict CAD alignments. Our results demonstrate that this way of first reconstructing an object-agnostic 3D scene representation and then performing object detection is more robust than frame-based methods [22,26].

Further, choosing an explicit 3D point cloud as an intermediate representation means that 3D reconstruction methods can be used out-of-the-box and can be applied in an online setting, unlike [35]. Applying FastCAD on the output of [18] our joint system can run online at 10 FPS (compared to less than 3 FPS [22]) while significantly improving the instance alignment accuracy on the Scan2CAD alignment benchmark from 43.0% [35] to 48.2%. Additionally, we introduce two metrics to assess the quality of retrieved shapes on the Scan2CAD [3] benchmark and show that FastCAD improves the introduced reconstruction accuracy from 22.9% [26] to 29.6%. In summary, our key contributions include:

- a novel and effective method for CAD model-based reconstruction where high-quality shape embeddings learned in a contrastive learning framework are distilled into an object detection network.
- an efficient system that predicts CAD retrievals and alignments for all objects in a scan in just 50 ms, allowing for online application to videos at 10 FPS.
- state-of-the-art alignment accuracy on the challenging and commonly used Scan2CAD benchmark for methods operating on scans (61.7% vs 61.2%) and on videos (48.2% vs. 43.0%).
- new evaluation metrics for the Scan2CAD benchmark assessing the quality of the retrieved shapes.

2 Related Work

Related work for this project comprises methods for *CAD retrieval and alignment*, *3D object detection* as well as general approaches for *CAD retrieval from an embedding space*.
Compared to generative methods that directly predict 3D shapes, retrieval-based methods have several advantages including guaranteeing realistic shapes with sharp edges and fine details. Generative methods on the other hand can struggle to make realistic and accurate predictions, particularly for unobserved object parts. The disadvantage of retrieval based methods is that they are constrained to represent those shapes that are available in the database. However, depending on the application, this limitation can be acceptable.

2.1 CAD Retrieval and Alignment

Using RGB-D scans as inputs. Methods like [2,4] attempt to retrieve CAD models for representing objects in an input point-cloud by first encoding the point-cloud into a feature volume and using predicted bounding boxes to crop parts of this feature volume. Subsequently, each crop is fed through a separate encoder to obtain shape embedding vectors. This is slower than our single-stage approach. To obtain CAD alignments [3,2,4] predict 3D correspondences for each object individually and then optimise for rotation and translation. [4] additionally predicts scene-layout elements and refines the positions of the CAD models to obey support relations in their scene graph. Their run times range from ca. 20 minutes [3] to 2.6 seconds [2,4]. Other methods [16,1] exhaustively render all CAD models in a database and optimise the pose of the best-fitting one by comparing rendered depth images to observed ones. However, with run times of more than 10 minutes per scene, these are not suited for real-time applications.
Using RGB videos as inputs. [22,26,35] predict CAD alignments from posed RGB videos. [22,26] both individually detect objects in each frame, associate them across frames, and perform a multi-view optimisation to find the best pose for each object. Approaches such as [22,26] are

very engineered and, due to the heterogeneity of their different modules, can usually not be trained end-to-end. This fact, in combination with a brittle tracking-by-detection step, makes them error-prone and unreliable. RayTran [35] does not perform per-frame predictions and instead relies on propagating the information into a 3D scene volume and performs predictions here. While doing so, they avoid the issues mentioned above, their mechanism for creating a 3D feature volume is computationally expensive with undisclosed run times and, in its current form, can not be run in an online setting.

2.2 3D Object Detection

3D object detection methods can be grouped by their underlying mechanism of aggregating per-object information. *Voting-based methods* [29,7,38,36] are initialised with a set of candidate object centres and require points to vote on whether they belong to a given object. Features of all points that voted to be part of the same object are aggregated and decoded to obtain a bounding box prediction. *Attention-based methods* [24,27] impose fewer inductive biases than voting-based methods. They replace the voting-based method of determining which features to aggregate with an attention-based method, resulting in softer assignments and alleviating the need for some hyper-parameters. *Convolution-based methods* [14,31,32] convert point clouds into voxels and process them with 3D convolutions. Densely processing features in 3D is very memory and compute-intensive. GSDN [14] improves the efficiency for such 3D convolution-based methods by introducing a generative sparse tensor decoder using a series of transposed convolutions and pruning layers. FCAF3D [31] used those transposed convolutions but introduced an anchor-free method that can better model the diversity of 3D object orientations and sizes. Simplifying the network architecture of [31] and introducing a multi-level object assigner [32] achieves a run-time of 21 FPS while further improving the performance. FastCAD uses the same backbone and neck as [32].

2.3 CAD Retrieval from an Embedding Space

Various previous works [28,11] have investigated learning an embedding space from which CAD models can be retrieved to model real-world objects. The most relevant of such works is [11], which learns a joint embedding space of noisy, incomplete scan objects and clean CAD models. They use 3D convolutions to learn feature embeddings for scans and CAD models. Their convolutional layers are trained by minimising a triplet loss [34] where they sample CAD models of a different category as negatives. Other works such as [23,19,20,21] learn CAD model embedding spaces by rendering CAD models and learning embeddings for the rendered and real images in a contrastive learning setting.

3 Method

FastCAD (Fig. 2) simultaneously predicts CAD alignments and shape embeddings for objects detected in a point cloud (Sec. 3.1). The predicted shape embeddings are used to retrieve the nearest CAD models from an embedding space. This embedding space is learned by encoding noisy, partial object scans and clean CAD models into a joint embedding space in a contrastive learning setting (Sec. 3.2).

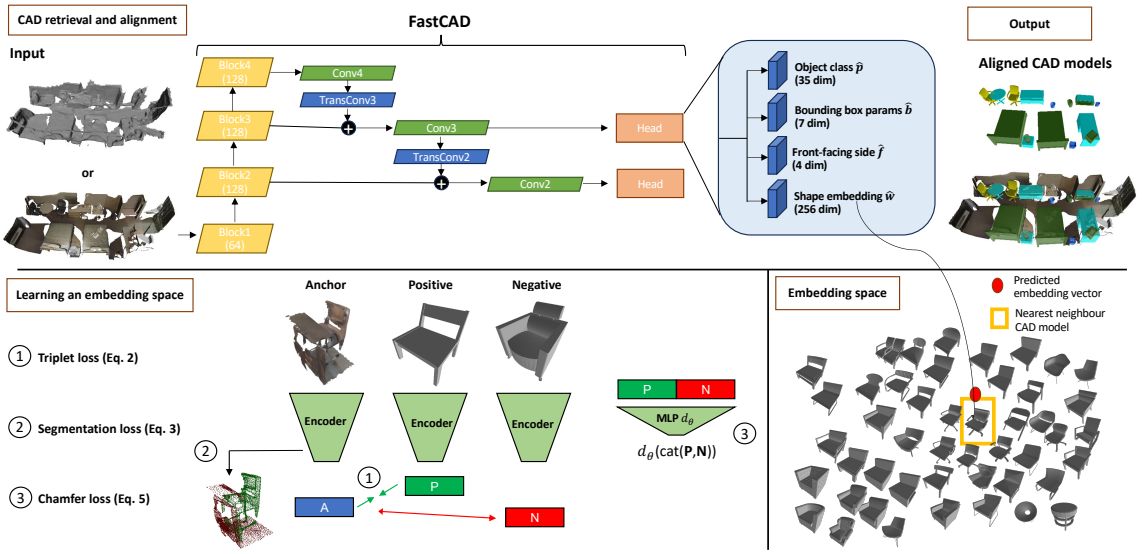

Fig. 2. Method. FastCAD retrieves and aligns CAD models for all objects detected in an input point cloud. For all detected objects it predicts their category $\hat{\boldsymbol{p}}$, bounding box parameters $\hat{\boldsymbol{b}}$, front-facing side $\hat{\boldsymbol{f}}$ and shape embedding $\hat{\boldsymbol{w}}$. The predicted embedding vector $\hat{\boldsymbol{w}}$ is used to retrieve the nearest neighbour CAD model from an embedding space previously learned in a contrastive learning setting with auxiliary tasks.

3.1 CAD Retrieval and Alignment

The input to FastCAD is a point cloud, which may be derived from (i) an RGB-D scan or (ii) a noisy scene reconstruction obtained, for example, by applying [33,18,5] to a video. This point cloud is encoded into a feature volume using a set of sparse 3D convolutions followed by generative transposed convolutions [15]. FastCAD's network architecture is inspired by [32]. For a range of sampled locations $(\hat{x}, \hat{y}, \hat{z})$ a shared detection head outputs classification probabilities $\hat{\boldsymbol{p}}$, oriented bounding box parameters $\hat{\boldsymbol{b}}$, front-facing side classification $\hat{\boldsymbol{f}}$ and shape embedding vector $\hat{\boldsymbol{w}}$. Depending on the average size of the predicted object class, the head output at feature level 2 or 3 is returned (level 2 for small objects, level 3 for large objects). For each oriented bounding box prediction $\hat{\boldsymbol{b}}$ we classify which of the four faces is the front face of the object using $\hat{\boldsymbol{f}}$. This information is used to choose between the four possible orientations when aligning the CAD model within the oriented bounding box. Encoding this information separately from the orientation in $\hat{\boldsymbol{b}}$ allows us to more easily leverage the symmetry annotations from Scan2CAD [3] which label each object to be non-symmetric or have 2-fold, 4-fold or complete rotational symmetry around the up-axis. For 2-fold, 4-fold and complete rotational symmetric objects, we modify the target front-facing side $\boldsymbol{f}$ from, e.g. $(1,0,0,0)$ to $\left(\frac{1}{2},0,\frac{1}{2},0\right)$, $\left(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right)$ and $\left(\frac{1}{4},\frac{1}{4},\frac{1}{4},\frac{1}{4}\right)$ respectively. This prevents the network from overfitting to arbitrary orientations for symmetric objects and allows it to generalise better (see Tab. 1). Through an assignment procedure, a detection i at $(\hat{x}, \hat{y}, \hat{z})$ may be matched with the nearest ground-truth object. This location then has ground-truth labels associated with it, and one

can formulate a loss function as:

$$\mathcal{L}_{\text{tot}} = \frac{1}{N_{\text{mat}}} \sum_{i=1}^{N_{\text{det}}} \mathcal{L}_{\text{cls}}(\hat{\boldsymbol{p}}_i, \boldsymbol{p}_i) + \mathbb{1}_i \left(\mathcal{L}_{\text{bb}}(\hat{\boldsymbol{b}}_i, \boldsymbol{b}_i) + \mathcal{L}_{\text{ff}}(\hat{\boldsymbol{f}}_i, \boldsymbol{f}_i) + \mathcal{L}_{\text{emb}}(\hat{\boldsymbol{w}}_i, \boldsymbol{w}_i) \right) \tag{1}$$

For each loss, predicted values are denoted with a hat. $\mathbb{1}_i = 1$ if detection i is matched to a ground-truth object and $\mathbb{1}_i = 0$ if not. $N_{\text{mat}} = \sum_{i=1}^{N_{\text{det}}} \mathbb{1}_i$ is the total number of matches. If a detection is not matched to a ground-truth object $\boldsymbol{p}_i = \mathbf{0}$. Note that each detection can be matched to only one ground-truth object, which is only matched if it is among the $k = 6$ closest detections to that ground-truth object. The classification loss $\mathcal{L}_{\text{cls}}$ is a focal loss, the bounding box loss $\mathcal{L}_{\text{bb}}$ is a DIoU loss [39], the front-facing side loss $\mathcal{L}_{\text{ff}}$ is a cross-entropy loss and the shape loss $\mathcal{L}_{\text{emb}}$ is a MSE loss. To obtain ground-truth shape embedding vectors $\boldsymbol{w}_i$ we first learn a CAD model embedding space (see Sec. 3.2).

3.2 Learned Embedding Space

We learn a shape embedding space using a contrastive learning setup with two new auxiliary tasks. For contrastive learning, we embed noisy object point clouds from scans and clean point clouds sampled from CAD models into a unified embedding space. For this purpose, we select all points within the Scan2CAD [3] object bounding boxes as anchor objects and associate the point clouds of the annotated CAD model as the positive example. We randomly sample different CAD models of the same category as negative examples. These three point clouds are passed through an encoder network to produce embedding vectors $\boldsymbol{w}$. We employ a triplet loss [34]:

$$\mathcal{L}_{\text{Triplet}} = \max(0, d^2(\mathbf{A}, \mathbf{P}) + m - d^2(\mathbf{A}, \mathbf{N})), \tag{2}$$

where $\mathbf{A}$, $\mathbf{P}$ and $\mathbf{N}$ are the embeddings of the anchor, positive and negative examples respectively. $d(\mathbf{A}, \mathbf{B})$ denotes the L2 distance between vector $\mathbf{A}$ and $\mathbf{B}$. This loss ensures that the distance between the anchor and the positive example is smaller by a margin m than the distance between the anchor and the negative. In addition to the triplet loss, we train the encoder to perform two auxiliary tasks. Doing so improves the quality of the retrieved shapes in FastCAD (see Tab. 2). The first task is to perform *foreground/background segmentation* of the input point clouds of the real scan. This is supervised with a binary cross-entropy loss:

$$\mathcal{L}_{\text{Segmentation}} = -\frac{1}{N_{\text{Seg}}} \sum_{i=1}^{N_{\text{Seg}}} (y_i \log(x_i) + (1 - y_i) \log(1 - x_i)) \tag{3}$$

Here $x_i \in [0, 1]$ are the predicted probabilities for each point, $y_i \in \{0, 1\}$ are the foreground/background labels and N_{Seg} is the number of points sampled. Note that we balance the ratio of foreground to background labels by only applying a loss to as many foreground points as there are background points. Otherwise, ca. 80%-90% of sampled points belong to the foreground class and we observe slightly smaller improvements to the quality of the embeddings.

For the second task, we train a shallow MLP, d_θ, to *regress the Chamfer distance* between the positive and the negative CAD model from their embeddings. The Chamfer distance $\mathrm{d}_{\text{Chamfer}}(X, Y)$ for point clouds X and Y is defined as

$$\mathrm{d}_{\text{Chamfer}}(X, Y) = \frac{1}{2} \left(\frac{1}{|X|} \sum_{x \in X} \min_{y \in Y} d(x, y) + \frac{1}{|Y|} \sum_{y \in Y} \min_{x \in X} d(x, y) \right) \tag{4}$$

The introduced loss is

$$\mathcal{L}_{\text{Chamfer}} = ||\mathrm{d}_\theta(\text{cat}(\mathbf{P}, \mathbf{N})) - \mathrm{d}_{\text{Chamfer}}(X_{\text{pos}}, X_{\text{neg}})||_1 \,, \tag{5}$$

where $\mathrm{d}_\theta(\text{cat}(\mathbf{P}, \mathbf{N}))$ is the Chamfer distance predicted from the concatenated embeddings $\mathbf{P}$ and $\mathbf{N}$ of the positive and negative CAD model. $\mathrm{d}_{\text{Chamfer}}(X_{\text{pos}}, X_{\text{neg}})$ is the ground-truth Chamfer distance computed using Eq. 4. The intuition behind introducing this loss is that sometimes the negative CAD model can be similar to the positive CAD model, while at other times, it may be very different. Forcing the encoder network to learn embeddings containing such information helps learn more useful embeddings. After training the encoder network, we compute embeddings for all CAD models in our training data. For each part of the scan that is annotated with a CAD model we then train FastCAD to predict the embedding vector $\hat{\boldsymbol{w}}$ associated with it and refer to this process as embedding distillation. At inference time for a given object detection and associated embedding prediction $\hat{\boldsymbol{w}}$ we retrieve the nearest neighbour CAD model of the predicted category $\hat{\boldsymbol{p}}$ and align it using the predicted bounding box $\hat{\boldsymbol{b}}$ and front-facing classification $\hat{\boldsymbol{f}}$.

4 Experimental Setup

4.1 Dataset

For training and testing our method, we use ScanNet [12] with CAD model annotations provided by Scan2CAD [3]. Those labels annotate the 1201 train scenes and 312 validation scenes from ScanNet [12] with CAD models from ShapeNet [6]. There are over 14K objects annotated with over 3K unique CAD models, which come from 35 categories, with the most popular categories being chair, table and cabinet.

4.2 Evaluation Metrics

For evaluating the CAD alignments, we follow the original evaluation protocol introduced by Scan2CAD [3]. A CAD model prediction is considered correct if the object class prediction is correct, the translation error is less than 20 cm, the rotation error is less than 20°, and the scale error is below 20%. For each scene and each category, as many predictions can be made as there are ground-truth CAD models. No duplicate predictions for the same ground-truth CAD model are considered.

Introducing reconstruction and shape accuracy metrics. The metric above does not evaluate the quality of the aligned shapes. To do so, we introduce the *Scan2CAD reconstruction accuracy*. For this metric, the individual checks on rotation, translation and scale are replaced by checking if the F-score at τ between the aligned predicted and aligned target CAD model is larger than a threshold μ and consider those a correct prediction. The F-score is defined as the harmonic mean of precision and recall, where precision is the fraction of points sampled on the predicted CAD model that lie within τ of points sampled on the ground-truth CAD model. Recall is the fraction of points on the ground-truth CAD within τ from a point on the predicted CAD model. Following [13], before computing the F-scores, objects are rescaled such that the largest side of the ground-truth CAD model has a length of 10 so that small and large objects are compared fairly. We set $\tau = 0.5$ and $\mu = 0.7$ (see the Supp. Mat. for results for different thresholds of μ). We also introduce the *Scan2CAD shape accuracy*, which follows the same protocol as the Scan2CAD reconstruction accuracy but computes the F-score when both the ground-truth and predicted CAD model are perfectly aligned, such as only focusing on the quality of the retrieved shape.

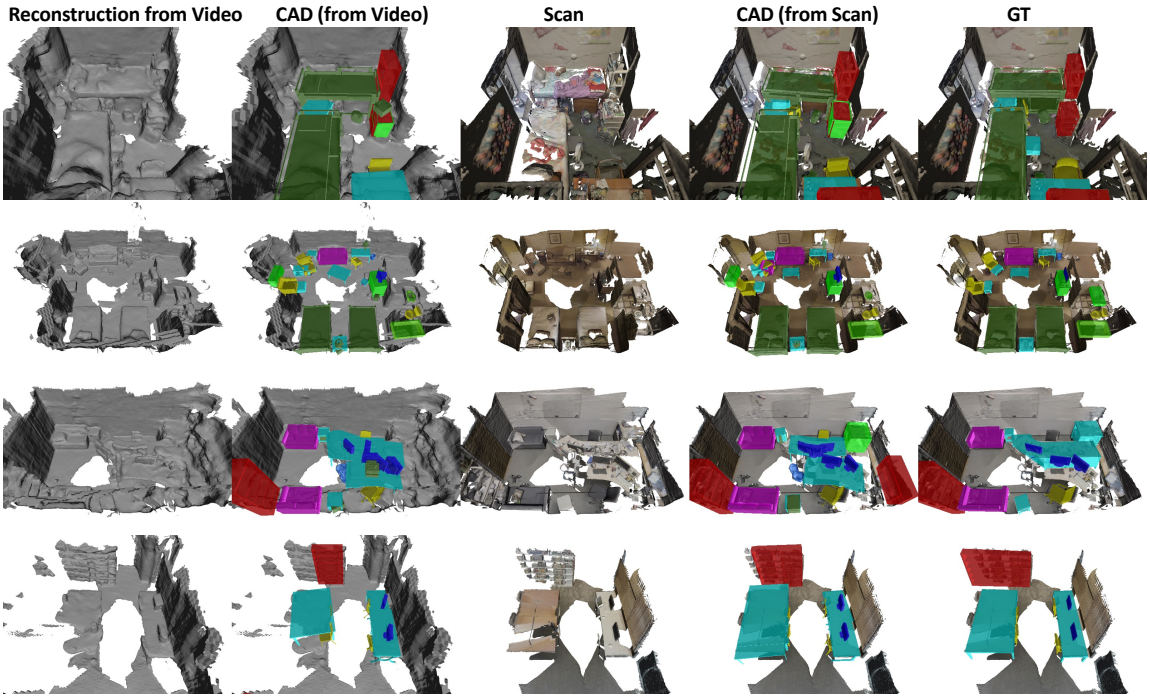

Fig. 3. Qualitative visualisation on ScanNet [12,3]. Column 1 shows the reconstruction generated by applying [18] to the input video. Column 2 shows the CAD retrieval and alignments predicted by FastCAD when operating on the reconstruction in column 1. Columns 3 and 4 show the input scan from ScanNet [12] and the CAD alignments FastCAD predicts for it. Column 5 shows the ground-truth CAD alignments from Scan2CAD [3].

4.3 Hyperparameters

For training the encoder network, we process all CAD models by normalising them to a unit cube and randomly sampling 1024 points from their surface. Similarly, cropped object scans are normalised and 1024 points are randomly sampled. Point clouds from cropped object scans with less than 1024 points are padded with 0s. We apply random scaling between 0.9 and 1.1, random translation between -0.1 and 0.1 and random rotation between $-10°$ and $10°$ on all point clouds. We use a Perceiver [17] as the encoder for the main experiments. It consists of three layers of cross-attention, each followed by two layers of self-attention, which share weights. The number of latent variables in the Perceiver [17] and their dimension is set to 256. The encoder network is trained for 750 epochs using a Lamb Optimiser [37] with a learning rate of 1e-3 and batch size of 25. The learned shape embeddings $\boldsymbol{w}$ also have dimension 256. The margin m in the triplet loss in Eq. 2 is set to 0.1. Foreground/background segmentation labels are predicted by cross-attending the final latent variables with the input point cloud. The MLP d_θ for predicting the Chamfer distance has a single hidden layer of size 256 and uses ReLU activation functions.
FastCAD is trained for 225 epochs using an AdamW optimiser [25] with a learning rate set to 1e-3 and weight decay by a factor of 10 after 120 and 165 epochs. Before processing each scene, the

Method	bathtub	bkshlf	cabinet	chair	display	sofa	other	table	trash bin	**class**	**instance**	time [ms]
Number of test instances #	120	212	260	1093	191	113	410	553	232	35	3184	-
Competing Methods - Input RGB-D Scan												
Scan2CAD [3]	36.2	36.4	34.0	44.3	17.9	30.7	**70.6**	30.1	20.6	35.6	31.7	740000
End-to-End CAD Retrieval [2]	38.9	41.5	51.5	73.0	26.5	76.9	26.8	48.2	18.2	44.6	50.7	2600
SceneCAD [4]	42.4	36.8	**58.3**	81.2	**50.7**	**82.9**	40.2	45.6	32.3	52.3	61.2	2600
Ours (Scan)	**43.3**	**47.2**	46.5	**85.7**	24.1	61.9	40.5	**56.1**	**69.8**	**52.8**	**61.7**	**50**
Competing Methods - Input RGB Video												
ODAM [22]	24.2	12.3	13.1	42.8	36.6	28.3	0.0	31.1	42.2	25.6	29.2	366
Vid2CAD [26]	28.3	12.3	23.8	64.6	**37.7**	26.5	6.6	28.9	47.8	30.7	38.6	3200
RayTran [35]	19.2	**34.4**	**36.2**	59.3	30.4	44.2	**27.8**	42.5	31.5	36.2	43.0	-
Ours (Video)	**35.0**	31.1	35.0	**71.5**	4.2	**54.0**	25.1	**48.8**	**48.7**	**39.3**	**48.2**	**100**
Ablations - Front-Facing Side Prediction												
Ours – discrete CAD orientation in embedding	27.5	36.3	42.7	85.5	**24.6**	61.1	33.4	47.7	50.0	45.4	56.2	50
Ours – front-facing side prediction	41.7	45.3	46.2	84.6	17.8	58.4	38.0	**56.8**	65.1	50.4	60.1	50
Ours – front-facing side prediction + symmetry	**43.3**	**47.2**	**46.5**	**85.7**	24.1	**61.9**	**40.5**	56.1	**69.8**	**52.8**	**61.7**	50

Table 1. Alignment accuracy on Scan2CAD [3] in comparison to the state-of-the-art. All numbers (except time and the first row) are percentages, and higher is better. FastCAD outperforms competing methods on scans and videos while dramatically reducing the inference time in both cases.

corresponding point cloud is down-sampled to a maximum of 50,000 points. During training, we perform a random sampling of input points (between 33% and 100%), random flipping along the x and y-axis with probability 50% as well as random rotation around the z-axis (from $-\pi$ to π), random scaling (between 0.9 and 1.1) and random translation (between -0.5 m and 0.5 m). Note that for predicting CAD alignments from videos, we train a separate version of FastCAD on the outputs of [18] for the training scenes in ScanNet [12]. This is because these more closely match the inputs that FastCAD receives at inference time for this setting.

4.4 Implementation Details

All code is implemented in PyTorch. FastCAD is integrated within the open-source object detection toolbox MMDetection3D [10]. It uses sparse convolutions from the Minkowski Engine [9,14]. Training on a single RTX 2080 takes ∼7 hours. Training the encoder network on the same GPU takes ∼24 hours.

5 Results

In Sec. 5.1 we evaluate our CAD alignments on the Scan2CAD alignment benchmark. In Sec. 5.2 we analyse the quality of the achieved reconstructions and the shape retrievals. Sec. 5.3 presents results when evaluating FastCAD's predictions continuously while reconstructing a scene. Finally, Sec. 5.4 ablates various components of the proposed system.

5.1 CAD Model Alignments

Comparing FastCAD to existing methods operating on RGB-D scans, we find that FastCAD performs similarly to SceneCAD [4], the previously most accurate method (61.7% vs. 61.2% instance alignment accuracy) while being more than 50 times faster (see Tab. 1). This massive speedup is mainly due to direct prediction of CAD alignments and shape embeddings in a single step. Compared to other methods when the input is an RGB video, FastCAD is not just significantly faster

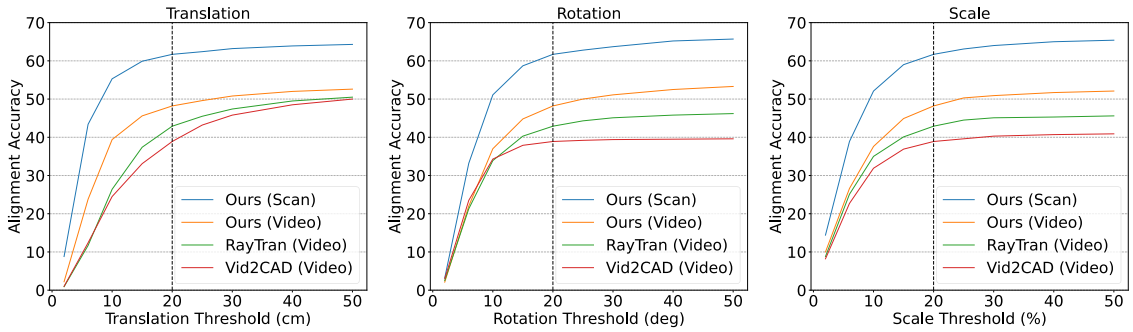

Fig. 4. Investigating different thresholds for the instance alignment accuracy on Scan2CAD [3]. The translation, rotation and scale thresholds, used to determine whether an alignment is correct, are varied from their default values at 20 cm, 20° and 20%. Note that in each plot, the thresholds that are not investigated remain at their default value. FastCAD outperforms competing methods across all thresholds.

but also considerably more accurate, outperforming the following best-competing method RayTran [35] by a large margin (48.2% vs. 43.0% alignment accuracy). We also compare our alignments to previous works at different thresholds for computing the alignment accuracy (see Fig. 4). Here, we find that we outperform them across all settings. Regarding run times, our total run-time to integrate new information from a new frame is just 100 ms (50 ms to run [18] plus 50 ms to run FastCAD on the reconstructed scene). This is significantly faster compared to ODAM [22] (366 ms), Vid2CAD [26] (3200 ms) and most likely also RayTran[3] [35] (see Fig 1).

We ablate our design decision for predicting the front-facing side $\hat{\boldsymbol{f}}$. In the first row of the last section in Tab. 1 we present the accuracies when encoding the information about the front-facing side of a CAD model in the shape embedding $\boldsymbol{w}$. In this case, each CAD model has four embedding vectors for each of the four discrete 90-degree orientations associated with it. At inference time the CAD model is aligned inside the predicted bounding box according to the discrete orientation of its nearest-neighbour embedding $\boldsymbol{w}$. The second row shows the accuracies when predicting the object front-facing side with an extra classification head (as explained in Sec. 3). This significantly improves the alignment accuracy (60.1% vs. 56.2%) while reducing the number of CAD embeddings that need to be stored and searched by a factor of four compared to the previous row. Finally, the last row shows that the alignment accuracy is further improved if the symmetry of the CAD model is taken into account when learning to predict the front-facing side (61.7% vs. 60.1%).

5.2 Reconstruction and Shape Quality

The CAD alignment accuracy used by [3,2,4,26,22,35] does not evaluate the quality of the retrieved CAD models. We therefore introduce two metrics, the *Scan2CAD reconstruction accuracy* and *Scan2CAD shape accuracy* as explained in Sec. 4. While the *Scan2CAD reconstruction accuracy* evaluates both the retrieved shapes and their alignments, the *Scan2CAD shape accuracy* only evaluates the quality of the retrieved shapes.
[3,2,4,35] do not have publicly available code and were not able to share their shape retrievals with

[3] See the Supp. Mat. for a discussion of this.

	Method	Alignment Acc.	Recon. Acc.	Shape Acc.	time [ms]
Competing Methods - Input RGB-D Scan					
Input RGB-D Scan	ScanNotate* [1]	78.2	60.1	83.5	660000
	Ours (Scan)	61.7	41.7	83.1	50
Competing Methods - Input RGB Video					
Input RGB Video	Vid2CAD* [26]	38.6	22.9	76.6	3200
	Ours (Video)	48.2	24.7	79.8	100
	Ours (Video, same retrieval Vid2CAD)	48.2	29.6	87.7	100
Ablation Experiments -Input RGB-D Scan					
Embedding Distillation	2-step retrieval: pred bbox	61.7	15.6	51.0	104
	2-step retrieval: nearest GT bbox	61.7	30.6	78.1	104
	Embedding distillation	61.7	41.7	83.1	50
Auxiliary Tasks for Training Encoder	Triplet	62.3	38.3	81.1	50
	Triplet + Chamfer	61.0	38.7	82.0	50
	Triplet + Segmentation	61.3	41.5	84.3	50
	Triplet + Chamfer + Segmentation	61.7	41.7	83.1	50
Encoder Architecture	PointNet++ [30]	61.5	29.6	74.0	50
	Perceiver [17]	62.3	38.3	81.1	50
Different Input Sources	ScanNet (Gray)	60.4	40.1	82.9	50
	DG Recon [18] (Gray)	48.2	24.7	79.8	100

Table 2. Alignment, reconstruction and shape accuracy on Scan2CAD [3] in comparison to competing methods and for various ablations. All accuracies are percentages and higher is better. Note that ScanNotate [1] initialises its CAD alignments from their ground-truth poses and Vid2CAD [26] constraints its CAD retrieval to the very small ground-truth scene pool, making some of their results not exactly comparable to ours.

us. We therefore compare our CAD retrievals to those from ScanNotate [1]. Note that ScanNotate [1] is used as an offline annotation method and optimises CAD retrievals and CAD poses, which are initialised from their ground-truth alignments[4]. While the alignment and reconstruction accuracy of ScanNotate [1] is better than ours (because the objects are initialised from ground-truth poses), we find that the shape accuracy, focusing only on the quality of the retrieved CAD model but not their alignment, is similar to ours. This is a significant achievement given that [1] exhaustively renders all CAD models in the database, leading to run-times that are more than four orders of magnitude larger than ours. In the video setting, FastCAD achieves better shape accuracies than Vid2CAD [26] even when Vid2CAD [26] limits its CAD retrievals to the ground-truth scene pool. When using the same retrieval setup as Vid2CAD [26], FastCAD achieves significantly better reconstruction accuracy (29.6% vs. 22.9%) and shape accuracy (87.7% vs. 76.6%).

To evaluate the quality of the embedding space, we compute the shape accuracy not just for the nearest neighbour retrieval, but also when retrieving instead the second, third or N-th nearest neighbour (see Fig 5). Here, we find that the shape accuracy remains high even when retrieving just the 10th closest CAD model. This demonstrates that geometrically similar CAD models are close to each other in the learned embedding space. This is desirable as it makes our CAD retrieval robust; even if the retrieved CAD model is not optimal, it will still closely match the observed object. Even retrieving the 100th closest CAD model from the learned embedding space is substantially more accurate than retrieving a random CAD model of the predicted category.

[4] We exclude ScanNotate predictions for those objects that FastCAD did not detect to partially mitigate the effect of ScanNotate having access to perfect object detections.

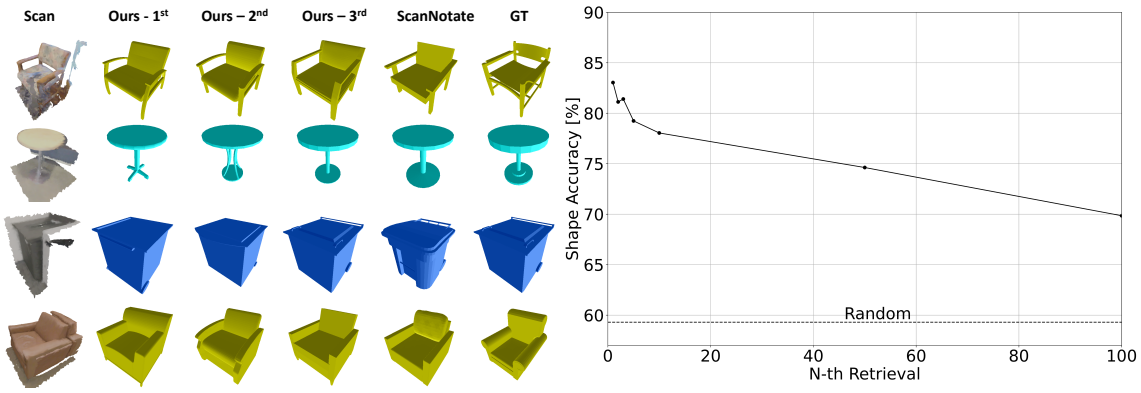

Fig. 5. CAD retrieval from the learned embedding space. Left: Qualitative visualisation of the retrieved CAD model for a given object in a scene. Note that the input to FastCAD from which a shape embedding $\hat{\boldsymbol{w}}$ is predicted is the scan of the entire scene. However, for clearer visualisation, we only show the cropped part of the scan for which a CAD model is retrieved. Across different object categories, our CAD retrievals are of similar high quality as the ones from the pseudo-labelling method ScanNotate [1] and the ground-truth CAD models from Scan2CAD [3]. Right: Our shape accuracy as a function of the N-th nearest CAD model retrieved from the embedding space. The shape accuracy remains high even as CAD models of increasingly worse rank are retrieved, which is a characteristic of a well-structured embedding space.

5.3 Incremental Evaluation for Online Setting

When predicting CAD alignment from an RGB video, we can evaluate the different metrics at various stages of the video sequence (see Fig 6). This is important for assessing the performance of our method for realistic applications in online settings, such as AR or robotics, where one requires not just an accurate final output but good performance throughout the sequence. Note that for computing the metrics, only those ground-truth CAD models whose centre has already appeared in the field of view of at least one seen frame are considered. Here, we find that the investigated metrics show good performance even early on (ca. 30% of the sequence). Nevertheless, we see a continuous improvement in all metrics as more parts of the video sequence are seen. We find that this is mainly because the output of the reconstruction method [18] improves continuously as parts of the scene that previously appeared far away are observed from up close. This improved quality of the scene mesh leads to more accurate CAD predictions by FastCAD. Another reason for the improvements over time is simply occlusion. Some centres of ground-truth objects may have appeared in the field of view but have not been observed as they were hidden behind other objects. This means that [18] can not reconstruct them, and consequently, FastCAD can not make predictions for those, reducing the accuracy.

5.4 Ablations

We ablate various design choices of FastCAD in the lower part of Tab. 2.
Embedding distillation. We investigate splitting bounding box detection and CAD retrieval into two successive steps. For this ablation, a detected bounding box is used to crop part of the input

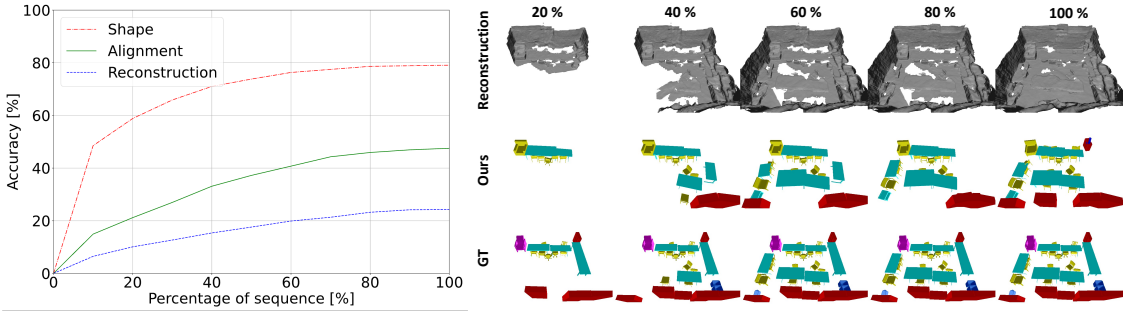

Fig. 6. Incremental evaluation of CAD predictions in an online setting. Left: Various metrics are investigated when only parts of the RGB video sequence have been seen. Right: Visualisation of the incremental evaluation for one scene. The reconstructed scene mesh, our CAD alignments and the considered ground-truth CAD alignments are visualised at various stages of the RGB video sequence.

point cloud, which then serves as the input to the encoder to produce the shape embedding. This is the only experiment where the encoder is used at test time. We observe poor reconstruction and shape accuracy (15.6% and 51.0%). This is due to a distribution shift in the input to the encoder, which was trained on object point clouds cropped with ground-truth bounding boxes but now receives point clouds cropped with predicted bounding boxes. However, even when using the nearest ground-truth bounding box to crop the input for the encoder, the final reconstruction accuracy and shape accuracy are significantly worse compared to using FastCAD to directly predict shape embeddings (41.7% vs 30.6% and 83.1% vs. 78.1%). We hypothesise that such notable improvements in the one-stage model are due to the significant mutual information between the shapes of the objects within the same environment, e.g. identical chairs around a table. The end-to-end shape embedding extraction, together with a large receptive field of the model, enables capturing such correlations.

Auxiliary tasks for training encoder. We analyse the effect of training our encoder network with the two proposed auxiliary tasks. Here, we observe improvements in the reconstruction accuracy and shape accuracy for training by predicting the Chamfer distance between the positive and the negative CAD model as well as performing foreground/background classification of the input point cloud (41.7% vs 38.3% and 83.1% vs. 81.1%). These metrics are computed from FastCAD, which was trained to regress the shape embeddings but not directly trained with the additional losses. Better training of the encoder leads to improved embeddings, which, even after distilling those into FastCAD, leads to notably better reconstruction and shape accuracies.

Encoder architecture. Testing different encoder architectures, we find that using a powerful encoder is crucial for obtaining high-quality shape embeddings. Compared to a standard PointNet++ [30] network, using a Perceiver [17] increases the reconstruction accuracy from 29.6% to 38.3% and the shape accuracy from 74.0% to 81.1%.

Different input sources. The output of [18] does not contain colour. To disentangle the effects of geometry and colour we input the point cloud from the RGB-D scan from ScanNet [12] without any colour information. Comparing the alignment, reconstruction and shape accuracy, we observe that while the significantly noisier inputs affect the performance, the achieved outputs are still of high quality (see also Fig. 3). Comparing the experiments for the RGB-D scans without colour information to the main experiment, we also find that colour adds only very little information, and

almost all information is contained in the geometry.

6 Conclusion

We propose FastCAD, which can retrieve and align CAD models to an input scene scan in just 50 ms due to its efficient design. By applying FastCAD to the output of online 3D reconstruction techniques, we can obtain precise CAD-model-based reconstruction from videos running in real-time at 10 FPS. We train and validate our system on Scan2CAD [3] which provides CAD model annotations for ScanNet [12]. Compared to competing works operating on scans, we reduce the runtime by a factor of 50 while slightly outperforming them regarding alignment accuracy. Compared to methods using videos as input, we improve the alignment accuracy from 43.0% to 48.2% while at least three times faster, thereby enabling real-time CAD-based reconstruction from videos. Despite those advances FastCAD is not free of errors. Typical errors include small misalignments of CAD models or over-detections of the same objects as different categories. These could be addressed through iterative refinement methods (at the cost of larger inference time) or by enforcing physicality constraints (e.g. plausible alignment with predicted scene lay-out elements like floor and walls and no 3D collisions between objects). Furthermore, in the online video setting FastCAD is applied anew to every updated input point-cloud. This can lead to some jittering of object shape and alignment. Future work could entail developing a mechanism to better ensure temporal consistency in this setting.

Acknowledgements

We would like to thank our colleagues at Qualcomm for their support and valuable inputs to the project. We also highly appreciate the constructive feedback from reviewers on improving the manuscript. Further, Florian wishes to express his sincere gratitude to his supervisors Ignas Budvytis and Roberto Cipolla for their invaluable guidance and support throughout his PhD studies.

References

1. Ainetter, S., Stekovic, S., Fraundorfer, F., Lepetit, V.: Automatically annotating indoor images with cad models via rgb-d scans. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (2023)
2. Avetisyan, A., Dai, A., Niessner, M.: End-to-end cad model retrieval and 9dof alignment in 3d scans. In: Int. Conf. Comput. Vis. (2019)
3. Avetisyan, A., Dahnert, M., Dai, A., Savva, M., Chang, A.X., Niessner, M.: Scan2cad: Learning cad model alignment in rgb-d scans. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019)
4. Avetisyan, A., Khanova, T., Choy, C., Dash, D., Dai, A., Nießner, M.: Scenecad: Predicting object alignments and layouts in rgb-d scans. In: Eur. Conf. Comput. Vis. (2020)
5. Bozic, A., Palafox, P., Thies, J., Dai, A., Nießner, M.: Transformerfusion: Monocular rgb scene reconstruction using transformers. Adv. Neural Inform. Process. Syst. (2021)
6. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
7. Cheng, B., Sheng, L., Shi, S., Yang, M., Xu, D.: Back-tracing representative points for voting-based 3d object detection in point clouds. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
8. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: IEEE Conf. Comput. Vis. Pattern Recog. (2020)
9. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. (2019)
10. Contributors, M.: MMDetection3D: OpenMMLab next-generation platform for general 3D object detection (2020)
11. Dahnert, M., Dai, A., Guibas, L., Nießner, M.: Joint embedding of 3d scan and cad objects. In: Int. Conf. Comput. Vis. (2019)
12. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: CVPR (2017)
13. Georgia Gkioxari, Jitendra Malik, J.J.: Mesh r-cnn. In: Int. Conf. Comput. Vis. (2019)
14. Gwak, J., Choy, C.B., Savarese, S.: Generative sparse detection networks for 3d single-shot object detection. In: Eur. Conf. Comput. Vis. (2020)
15. Gwak, J., Choy, C.B., Savarese, S.: Generative sparse detection networks for 3d single-shot object detection. In: Eur. Conf. Comput. Vis. (2020)
16. Hampali, S., Stekovic, S., Sarkar, S.D., Kumar, C.S., Fraundorfer, F., Lepetit, V.: Monte carlo scene search for 3d scene understanding. In: IEEE Conf. Comput. Vis. Pattern Recog. (2021)
17. Jaegle, A., Gimeno, F., Brock, A., Zisserman, A., Vinyals, O., Carreira, J.: Perceiver: General perception with iterative attention (2021)
18. Ju, J., Tseng, C.W., Bailo, O., Dikov, G., Ghafoorian, M.: Dg-recon: Depth-guided neural 3d scene reconstruction. In: Int. Conf. Comput. Vis. (2023)
19. Kuo, W., Angelova, A., Lin, T.Y., Dai, A.: Mask2cad: 3d shape prediction by learning to segment and retrieve. In: Eur. Conf. Comput. Vis. (2020)
20. Kuo, W., Angelova, A., Lin, T.Y., Dai, A.: Patch2cad: Patchwise embedding learning for in-the-wild shape retrieval from a single image. Int. Conf. Comput. Vis. (2021)
21. Langer, F., Budvytis, I., Cipolla, R.: Leveraging geometry for shape estimation from a single rgb image. In: Brit. Mach. Vis. Conf. (2021)
22. Li, K., DeTone, D., Chen, S., Vo, M., Reid, I., Rezatofighi, H., Sweeney, C., Straub, J., Newcombe, R.: Odam: Object detection, association, and mapping using posed rgb video. In: Int. Conf. Comput. Vis. (2021)
23. Li, Y., Su, H., Qi, C.R., Fish, N., Cohen-Or, D., Guibas, L.J.: Joint embeddings of shapes and images via cnn image purification. ACM Trans. Graph. (2015)

24. Liu, Z., Zhang, Z., Cao, Y., Hu, H., Tong, X.: Group-free 3d object detection via transformers. In: Int. Conf. Comput. Vis. (2021)
25. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: Int. Conf. Learn. Represent. (2019)
26. Maninis, K.K., Popov, S., Nießner, M., Ferrari, V.: Vid2cad: Cad model alignment using multi-view constraints from videos. IEEE Transactions on Pattern Analysis and Machine Inttelligence (2022)
27. Misra, I., Girdhar, R., Joulin, A.: An End-to-End Transformer Model for 3D Object Detection. In: Int. Conf. Comput. Vis. (2021)
28. Pham, Q.H., Tran, M.K., Li, W., Xiang, S., Zhou, H., Nie, W., Liu, A., Su, Y., Tran, M.T., Bui, N.M., Do, T.L., Ninh, T.V., Le, T.K., Dao, A.V., Nguyen, V.T., Do, M.N., Duong, A.D., Hua, B.S., Yu, L.F., Nguyen, D.T., Yeung, S.K.: RGB-D Object-to-CAD Retrieval. In: Eurographics Workshop on 3D Object Retrieval (2018)
29. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: Int. Conf. Comput. Vis. (2019)
30. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Adv. Neural Inform. Process. Syst. (2017)
31. Rukhovich, D., Vorontsova, A., Konushin, A.: Fcaf3d: fully convolutional anchor-free 3d object detection. In: Eur. Conf. Comput. Vis. (2022)
32. Rukhovich, D., Vorontsova, A., Konushin, A.: Tr3d: Towards real-time indoor 3d object detection (2023)
33. Sayed, M., Gibson, J., Watson, J., Prisacariu, V., Firman, M., Godard, C.: Simplerecon: 3d reconstruction without 3d convolutions. In: Eur. Conf. Comput. Vis. (2022)
34. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: IEEE Conf. Comput. Vis. Pattern Recog. (2015)
35. Tyszkiewicz, M.J., Maninis, K.K., Popov, S., Ferrari, V.: Raytran: 3d pose estimation and shape reconstruction of multiple objects from videos with ray-traced transformers. In: Eur. Conf. Comput. Vis. (2022)
36. Wang, H., Shi, S., Yang, Z., Fang, R., Qian, Q., Li, H., Schiele, B., Wang, L.: Rbgnet: Ray-based grouping for 3d object detection. In: IEEE Conf. Comput. Vis. Pattern Recog. (2022)
37. You, Y., Li, J., Hseu, J., Song, X., Demmel, J., Hsieh, C.: Reducing BERT pre-training time from 3 days to 76 minutes. Int. Conf. Learn. Represent. (2020)
38. Zhang, Z., Sun, B., Yang, H., Huang, Q.X.: H3dnet: 3d object detection using hybrid geometric primitives. In: Eur. Conf. Comput. Vis. (2020)
39. Zheng, Z., Wang, P., Liu, W., Li, J., Ye, R., Ren, D.: Distance-iou loss: Faster and better learning for bounding box regression. In: AAAI (2020)