Patch2CAD: Patchwise Embedding Learning for In-the-Wild Shape Retrieval from a Single Image

Weicheng Kuo¹, Anelia Angelova¹, Tsung-Yi Lin¹, Angela Dai²
¹ Google Research, Brain Team
² Technical University of Munich
{weicheng, anelia, tsungyi}@google.com, angela.dai@tum.de

Figure 1: From an input RGB image, we learn a shared image-CAD embedding space by embedding patches of detected objects from the RGB images and patches of CAD models. By establishing patch-wise correspondence between image and CAD, we can establish object correspondence based on part similarities, enabling more effective shape retrieval for new views, as well as robust top-k CAD retrieval. Our patch-based retrieval for a similar 3D CAD representation coupled with pose prediction results in a CAD-based 3D understanding of the objects in the image.

Abstract

3D perception of object shapes from RGB image input is fundamental towards semantic scene understanding, grounding image-based perception in our spatially 3-dimensional real-world environments. To achieve a mapping between image views of objects and 3D shapes, we leverage CAD model priors from existing large-scale databases, and propose a novel approach towards constructing a joint embedding space between 2D images and 3D CAD models in a patch-wise fashion – establishing correspondences between patches of an image view of an object and patches of CAD geometry. This enables part similarity reasoning for retrieving similar CADs to a new image view without exact matches in the database. Our patch embedding provides more robust CAD retrieval for shape estimation in our end-to-end estimation of CAD model shape and pose for detected objects in a single input image. Experiments on in-the-wild, complex imagery from ScanNet show that our approach is more robust than state of the art in real-world scenarios without any exact CAD matches.

1. Introduction

Fundamental to many visual perception tasks is an understanding of the decomposition of an observed scene into its constituent objects, and the semantic meaning of these objects – including class categorization, segmentation, and structural 3D understanding. In recent years, advances in 2D object recognition and localization have achieved impressive success in image-based understanding, even from only single image input [22, 18, 46, 21]. Such recognition and perception constrained to the image domain unfortunately remains limited towards understanding 3D attributes.
such as shape and structure, which are not only fundamental towards a comprehensive, human-like understanding of objects in a scene, but towards many applications such as autonomous exploration and interaction of an environment.

To address 3D perception from a single RGB image, we have recently seen several methods proposed taking a generative approach towards reconstructing the observed objects’ geometry [19, 42, 14]. These methods show promising results in attaining 3D understanding of objects in complex scene imagery, but take a low-level approach towards geometric reconstruction, constructing voxel-by-voxel or vertex-by-vertex, often resulting in noisy or oversmoothed geometry, or geometry not representing a valid object instance (e.g., missing a leg on a chair). In contrast, several approaches have taken a CAD-based prior for representing the 3D structure of the objects seen in an RGB or RGB-D observation, by retrieving and aligning CAD models from a database similar to the observed objects [41, 33, 2, 30]. This CAD prior enables representation of each object with a clean, complete 3D mesh known to represent valid instances of objects and able to be stored compactly for potential downstream applications. Unfortunately, such a retrieval-based approach tends to struggle with generalization, in particular when a new observed image of an object does not exactly match any CAD model in the dataset.

We observe that in these challenging scenarios, various part similarities can be leveraged to find a similar shape. Thus, we propose Patch2CAD, which constructs a joint embedding space between images and CAD models based on encoding mid-level geometric relations by establishing similarity of patches of images to patches of object geometry. These correspondences can be aggregated into CAD prediction by majority voting. This enables CAD retrieval based on the predominant part similarities, enabling improved generalizability for CAD retrieval to reconstruct the shapes of objects seen in an image.

To achieve a 3D understanding of object structure from a single RGB image, we first detect object locations in 2D, then construct our patch-based image-CAD embedding space enabling voting for retrieval of a similar CAD model, and predict the pose of that CAD in the image. Patch2CAD is trained end-to-end to comprehensively establish an effective image-CAD embedding.

Our main contribution is a patch-based learning of a joint embedding space between the two very different domains of 2D images and 3D CAD models, which establishes more robust, part-level-based correspondences (see Figure 1). We demonstrate that this patch-wise embedding enables meaningful CAD retrievals for image observations not just in the top nearest neighbor, but for top-k retrieval. As a result, we achieve more effective association of CAD shapes to images observations of objects with no exact CAD matches in a candidate database, as is typically the case for real-world scenarios. We demonstrate Patch2CAD’s effective 3D shape perception on both ScanNet [11] and Pix3D [52] datasets. In particular, on the complex, in-the-wild images from ScanNet [11], Patch2CAD exhibits notable advantage to its patch-based approach, outperforming state of the art by 1.9 Mesh AP (22% relative improvement).

2. Related Work

Scene understanding is one of the fundamental problems in computer vision. A vast amount of literature on the topic has forwarded the field in understanding of 2D images: for example popular methods for object detection [18, 46, 45, 37, 36, 32, 15, 58], semantic segmentation [38, 23] and instance segmentation [21, 31]. Our approach is inspired by these 2D image understanding approaches, but instead focuses on producing a 3D representation of the objects observed in a single image, providing additional geometric, structural information about the scene.

Single-View Object Reconstruction. Recently, we have also seen remarkable progress in reconstructing the 3D shape of an object from a single RGB image. Such work has also been driven by shape representations: earlier research focused on dense volumetric grids [10, 57], while point clouds [16, 59] and hierarchical structures such as octrees [53, 47] offered more memory and computationally efficient representations. Mesh-based approaches offer an efficient surface representation along with adaptive structures, but tend to rely on strong topological assumptions, taking a deformation-based approach from a given template mesh [55, 56]; generative approaches without relying on templates tend to be limited to small numbers of vertices [12]. Implicit functions have recently seen notable success in single-object shape reconstruction, characterizing shape by prediction an occupancy or signed distance field value for a location in space [40, 44, 49]. Approaches to predict convex primitives have also been shown to produce promising results [13].

While these approaches operated on images encompassing only one object, Mesh R-CNN pioneered an approach for generating the shapes for multiple objects seen in an RGB image, which more closely represents real-world perception scenarios. Several methods have now furthered development on this task; Mask2CAD [30] proposed a CAD retrieval-based approach towards understanding the shape and pose of objects, and Nie et al. [42] a mesh generation approach for object reconstruction based on initial deformation from a sphere followed by edge refinement to handle local topology errors. Our approach also tackles shape reconstruction for the multiple objects seen in an RGB image, leveraging CAD retrieval and focusing on the construction of a robust image-CAD embedding space.
CAD-Based Retrieval and Alignment. An alternative to generative methods for reconstruction is to leverage CAD model priors to represent objects in a scene, and retrieve and align them to achieve a scene reconstruction composed of clean, compact mesh representations of each object. Early work in computer vision demonstrated the use of existing geometric models as priors [6, 9, 48]; the current availability of large-scale CAD model datasets (e.g., ShapeNet [7], Pix3D [52]) has revitalized this approach. Various methods have been introduced for CAD model retrieval and alignment to RGB-D scans [50, 29, 33, 5, 20, 27], including end-to-end learning pipelines [3, 4], as well as CAD alignment to an image assuming that shape is given [35, 17, 24].

From a single image, Aubry et al. [1] develop hand-crafted HOG-based features to match textured renderings of CAD models to images in order detect chairs; our approach learns to associate CAD patches with image patches based on a more general learning of purely geometric correspondence, enabling learning geometric structures as well as the use of geometry-only CAD databases. More recently, Izadinia and Seitz [28] and Huang et al. [25] apply analysis-by-synthesis approaches for CAD model alignment and scene layout estimation from a single image, leveraging a costly optimization (minutes to hours) for each input image.

Shape retrieval methods also show promising results by learning joint RGB-CAD space embedding [54, 34, 39, 30]. Li et al. [34] propose a method to construct a joint embedding space between RGB images and CAD models, enabling CAD model retrieval from images; the embedding space is first constructed from shape descriptors, and then image embeddings are optimized for into the shape space. Massa et al. [39] learns to adapt object RGB features to CAD space with a projection layer for object instance detection. Kuo et al. [30] jointly optimize for a shared embedding space between image views and CAD models in order to perform retrieval for multiple objects seen in an image. Such techniques can be prone to overfitting, as an object shape obtains a single, global representation, and a new image may not contain exact CAD matches but rather various part similarities. Our approach addresses a similar problem statement in learning a mapping between images and CAD models; however, to better generalize to new observations with inexact matches, leveraging a majority part similarity to more robustly retrieve CAD models for reconstruction.

3. Method

3.1. Overview

From a single input RGB image, we aim to understand the observed scene by predicting the object semantics and 3D structures, by retrieving and aligning similar CAD models to the observed image. Objects are first detected in the 2D image, represented by their 2D bounding box, class label, and 2D instance segmentation mask. We then aim to learn a shared embedding space between the image representation of the objects and CAD models, in order to retrieve a similar CAD model representing the 3D structure of a detected object. In a separate pose-prediction head, we simultaneously regress the pose of the CAD model that aligns it with the image observation.

A shared embedding space between image and CAD can be difficult to effectively construct due to the strong differences between the two domains. While mapping image observations of an object together with the full CAD models into a shared embedding space has shown promise [34, 30], this approach tends to struggle with generalization to views of objects without exact matches in the CAD database. Thus, rather than constructing an embedding space which maps similar image observations of an object together with full CAD models, we aim to learn an embedding space which captures not only global semantic similarity between image and CAD, but mid-level and low-level geometric similarities. We propose to learn the embedding of object parts and CAD parts by constructing a shared feature space where patches of object images lie close to similar patches of CAD objects. This enables reasoning about similar parts for retrieval in a scenario without exact CAD matches for a new image view, enabling more robust CAD reconstruction.

3.2. 2D Object Detection

We leverage a state-of-the-art 2D object detection and instance segmentation backbone to inform our 3D shape reasoning. From an input RGB image, 2D object bounding boxes and class labels are localized using RetinaNet [36], and instance segmentation masks are predicted using ShapeMask [31]. The learned features from the 2D object detection guide our shape prediction; for a detected object $k$, we use the predicted box for an object to crop features $f_k$ corresponding to the object, and multiply with the instance mask prediction $m_k$. We then use $m_k \circ f_k$ as input for the image-shape embedding as well as pose alignment.

3.3. Patch-Based Joint Embedding Learning

Our approach centers around constructing a patch-based joint embedding space between the two domains of image observations of an object and 3D CAD model representations of objects. While humans can establish perceptual correspondence between images and CAD models, it is challenging to bridge the domains due to strong differences in representation: in contrast to a 3D geometric CAD model, an image is view-dependent, colored, and contains lighting and material effects. Moreover, in real-world scenarios, we typically do not have exact CAD matches to the image views as groundtruth annotation. We construct an embedding space between image patches and CAD patches, to enable reasoning about mid-level and low-level structural
Figure 2: Our goal is to learn a shape embedding space for retrieval by leveraging patch correspondence between RGB and shape. At train time, we sample RGB patches from object regions and rendered shape patches from the object class. We shape the embedding space with a contrastive loss, and regularize the learning with surface normal matching so that positive patches have high geometric similarity, while negative patches come from non-matching shapes with low geometric patch similarity. This patch-wise construction establishes more robust correspondence for shape retrieval from images at test time.

<table>
<thead>
<tr>
<th>Top Retrieved Shape Patches</th>
<th>Top Retrieved Shape Patches</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Top Retrieved Shape Patches" /></td>
<td><img src="image2" alt="Top Retrieved Shape Patches" /></td>
</tr>
</tbody>
</table>

\[ L_e = - \sum_{a \in A} \log \frac{D_p(a)}{D_p(a) + CD_n(a)} \]  
\[ D_p(a) = \frac{1}{|P(a)|} \sum_{p \in P(a)} \exp(D(f_{im}^a, f_p^c)) \]  
\[ D_n(a) = \frac{1}{|N(a)|} \sum_{n \in N(a)} \exp(D(f_{im}^a, f_n^c)) \]

where \( L_e \) denotes the total loss, \( A \) the set of all anchor (query) patches, \( P(a) \) and \( N(a) \) the positive and negative matches for the query patch \( a \), \( C = 24 \) a weighting value, \( D(x, y) = (x/||x||)_T(y/||y||)/\tau \) with \( \tau = 0.15 \). \( D_p(a) \) and \( D_n(a) \) are the mean exponentiated weights of positive and negative pairs. To further improve learning efficiency, we exclude empty RGB and shape patches in the embedding loss, as determined by the rendered binary masks.

Our overall loss is similar to Mask2CAD, but because we operate on patch-level correspondence, we removed hard-positive mining due to relaxed constraints on patch matching vs. exact instance matches. The formulation of our loss is different from standard InfoNCE loss because we have multiple positives (shape rendering patch) for each query patch. Thus, we need to balance the ratio of positives/negatives per batch via the \( C \) parameter.

**Patch similarity for embedding construction.** To train our embedding construction, we establish patch similarity for matching and non-matching patches between image and CAD patches by estimating their geometric similarity. We use rendered normals from the CAD models (with normals represented in canonical space) and their corresponding patches to represent local geometry, and for the image, we use patches of the rendered normals of the ground truth corresponding CAD model. For each patch of normals, we...
compute its descriptor by a self-similarity histogram, evaluated as the histogram over all pairwise angular distances of the normals in the patch; histograms are normalized to sum to 1. This allows us to estimate orientation-independent geometric similarity. We then measure the difference between two patches of normals by IoU of their self-similarity histograms. Positive matches to a query are determined by patches corresponding to a ground truth CAD annotation with self-similarity IoU $> \theta_p$, and negative matches by patches corresponding to non-corresponding CAD models with self-similarity IoU $< \theta_n$. Since a ground truth CAD annotation may contain patches that are dissimilar to the query patch and non-ground truth CAD models may contain patches that are similar to the query, we empirically found that double thresholds helped to avoid such associations. We set $\theta_p, \theta_n = 0.4, 0.6$ in our implementation.

We additionally employ hard negative mining by sampling the top negative patches by distance to the query. During training, we take $16 \times$ the number of objects per image for hard negative examples. This enables for better distinguishing on difficult cases, and an improved embedding space. We set $|N(a)| = 1024$ for each anchor patch. Regarding hard-positive mining, we observe that it hurts the performance with a fixed top-$K$ mining due to unstable number of positive pairs per batch. To remedy this, we average the weights of all positive pairs and treat them as one positive sample $D_q(a)$, which leads to significantly more stable learning and better performance.

### 3.4. Patch-Based Retrieval

Since our joint embedding of images and shapes is constructed patch-wise, we can leverage many patch retrievals for a more robust, comprehensive shape retrieval. We use randomly sampled patches from CAD renderings to construct the database for retrieval. Then for a detected object in an image, we randomly sample $K_q$ patches from $f^m$, and for each patch, we retrieve $K_r$ patches from the database. The $K_r$ retrieved patches are then used to decide the corresponding CAD model of the patch query by majority voting, resulting in $K_q$ CAD models for each patch; the final shape retrieval is obtained by majority vote of the $K_q$ patch-retrieved CADs, excluding those retrieved by patches fully outside the predicted instance mask. While an image-CAD mapping based on full image view of the object and whole CAD model might struggle to retrieve from a global similarity perspective under inexact matches, our patch-based shape retrieval encourages the retrieved shapes to more comprehensively match the image.

### 3.5. Pose Prediction

We simultaneously predict the pose of the 3D shape corresponding to its 2D image observation in a separate branch. Similar to [30], we predict the rotation of the shape by a rotation classification followed by a regression refinement, and the translation as an offset from the 2D bounding box center. To obtain the estimated rotation, use rotation bins computed by K-medoid clustering of the train object rotations as quaternions, and predict the bin using a cross entropy loss, followed by predicting a refinement offset quaternion with a Huber loss [26]. Translation is estimated as an offset from the predicted bounding box center as a ratio of the box dimensions, and optimized with a Huber loss.

### 3.6. Implementation Details

Our ShapeMask [31] instance segmentation backbone (ResNet-50-FPN) is initialized with COCO pretraining, and our embedding for both image and CAD renderings uses a ResNet-18-FPN backbone with random initialization. We train our instance segmentation for amodal bounding box prediction instead of modal boxes in the standard COCO setup, as this can capture more consistent context and provides more stable guidance for the pose translation estimation. We additionally apply data augmentation to improve generalization, including HSV color jitter, ROI box jitter, and image scale jitter during training.

We train our approach for 36K iterations using a batch size of 256 on ScanNet, which takes $\approx 2$ days. The learning rate is initialized to 0.16 and decreased by $10x$ at 24K and another $10x$ at 30K iterations. In terms of inference time, Patch2CAD takes $\approx 74ms$ per image $^1$, 58ms model + 16ms retrieval (vs. Mask2CAD $\approx 60ms$), with unoptimized parallel patch retrievals.

### 4. Experiments

We evaluate our approach on the ScanNet dataset [11], which contains challenging real-world imagery of multiple objects per image in cluttered indoor environments, with many occlusions, partial views and varying lighting conditions. The ScanNet dataset contains 1513 indoor scenes; we use the Scan2CAD [2] annotations of ShapeNet [7] CAD models to the ScanNet scenes to provide ground truth CAD correspondences for training and evaluation. Note that there are no exact matches between the CAD models to the real-world imagery, as is reflective of many real-world application scenarios. Following the Mask2CAD [30] evaluation protocol on ScanNet, we use the 25K frame subset provided by the dataset for training and validation, containing 19387 train and 5436 validation images, respectively.

In addition, we evaluate our approach on the Pix3D dataset [52], which contains 10,069 images of indoor furnishings labeled with corresponding CAD models. We use the train/test split by Mesh R-CNN [19] for direct comparison.

---

$^1$Measured on Pix3D for comparison with Mask2CAD [30].
Table 1: Performance on ScanNet [11]. We report mean $\text{AP}_{\text{mesh}}$ and per-category $\text{AP}_{\text{mesh}}$.

<table>
<thead>
<tr>
<th>ScanNet 25K</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask2CAD [30]</td>
<td>8.4</td>
<td>23.1</td>
<td>4.9</td>
</tr>
<tr>
<td>Patch2CAD (Ours)</td>
<td><strong>10.3</strong></td>
<td><strong>26.0</strong></td>
<td><strong>6.6</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bed</th>
<th>sofa</th>
<th>chair</th>
<th>cabinet</th>
<th>trashbin</th>
<th>display</th>
<th>table</th>
<th>bookshelf</th>
</tr>
</thead>
<tbody>
<tr>
<td>14.2</td>
<td>13.0</td>
<td>13.2</td>
<td>7.5</td>
<td>7.8</td>
<td>5.9</td>
<td>2.9</td>
<td>3.1</td>
</tr>
<tr>
<td>18.8</td>
<td>12.4</td>
<td>17.6</td>
<td>7.5</td>
<td>8.6</td>
<td><strong>10.8</strong></td>
<td><strong>3.3</strong></td>
<td><strong>3.3</strong></td>
</tr>
</tbody>
</table>

**Figure 3:** Qualitative results on ScanNet [11] images, in comparison with state-of-the-art Total3D [42] and Mask2CAD [30]. Our patch-based shape embedding results in more accurate shape retrieval as well as more robust retrieval for strongly occluded objects (see rows 3, 4, 9, 10). Note that different colors denote distinct object instances in the visualization.

**Evaluation metrics.** We adopt previously established metrics for both 2D and 3D evaluation. For evaluating 2D outputs, we employ the predominant metrics from 2D object recognition: $\text{APbox}$ and $\text{APmask}$ on the 2D detections.
Table 2: Performance on ScanNet [11] using groundtruth 2D detections. We report mean AP\textsuperscript{mesh} and per-category AP\textsuperscript{mesh}.

<table>
<thead>
<tr>
<th>Method</th>
<th>Patch Size</th>
<th>Normals</th>
<th>AP</th>
<th>AP50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mask2CAD [30]</td>
<td>1.0</td>
<td>V</td>
<td>8.4</td>
<td>23.1</td>
</tr>
<tr>
<td>Patch2CAD (Ours)</td>
<td>1.0</td>
<td>V</td>
<td>9.4</td>
<td>24.7</td>
</tr>
<tr>
<td>Patch2CAD (Ours)</td>
<td>0.5</td>
<td>V</td>
<td>9.5</td>
<td>24.5</td>
</tr>
<tr>
<td>Patch2CAD (Ours)</td>
<td>0.33</td>
<td>V</td>
<td>10.3</td>
<td>26.0</td>
</tr>
<tr>
<td>Patch2CAD (Ours)</td>
<td>0.25</td>
<td>V</td>
<td>10.0</td>
<td>25.8</td>
</tr>
</tbody>
</table>

Table 3: Performance vs patch size and use of normals. 1.0 patch size corresponds to the full object size.

What is the effect of patch-wise embedding learning on representing shape geometry? In Table 1, we see that our patch-based embedding improves a retrieval-based 3D object reconstruction, in contrast to the whole-shape embedding of Mask2CAD. We additionally evaluate our patch-based embedding for retrieval given ground truth 2D detections in Table 2, showing consistent improvement over both the Mask2CAD retrieval and the mesh generation approach of Nie et al. [42]. Note that we use the training scheme of Nie et al. on SUN RGB-D [51], as they use scene layout information during training (similar to ScanNet, SUN RGB-D is also captured from real indoor scenes with a PrimeSense-based sensor). Finally, we evaluate Patch2CAD given ground truth 2D detections as well as pose (i.e., evaluating shape only) in comparison with Nie et al. [42] as well as Mask2CAD [30] in Table 4, using an F-score for shape reconstruction evaluation. Even with a

detail of objects. We use the newly introduced AP\textsuperscript{mesh} metric [19] to evaluate the 3D shape and pose predictions for the 3D objects. Similar to Mask2CAD [30], we evaluate AP\textsuperscript{mesh} metrics at IoU 0.5 (AP50) at IoU 0.75 (AP75), as well as AP as the mean over AP50-AP95, using 10 IoU thresholds between 0.5 and 0.95. For more consistent reproducibility, we report our evaluations as an average of 2 independent runs. The thresholds used in F-scores follow [30] on ScanNet and [19, 30] on Pix3D.

Comparison to the state of the art. In Table 1, we evaluate our 3D object understanding from a single image in comparison to Mask2CAD [30] on the ScanNet [11] benchmark proposed by Mask2CAD. Our improvement on Mesh AP50 is larger than AP75, showing that Patch2CAD maintains more robust shape estimate even when it retrieves non-exact matches.

Figure 4: Top-\(k\) nearest neighbor retrievals from detected objects in ScanNet [11] images with Scan2CAD [2] ground truth CADs, in comparison to Mask2CAD [30] (same detection inputs). Our approach achieves a more consistent shape embedding space, enabling robust top-\(k\) retrieval with structurally similar CAD associations for not only the top-1 nearest neighbor.
shape-only prediction, strong occlusions in the image views can be challenging; Patch2CAD maintains more robustness.

**Effect of patch size and use of surface normals.** Table 3 analyzes various patch sizes and with/without surface normals. The first row corresponds to Mask2CAD; ours on the fourth row. Our $\frac{3}{4}$ patch basis and use of normals helps to notably improve shape retrieval.

**Comparison on Pix3D.** Table 5 shows a comparison with both generative and retrieval methods on Pix3D [52]. Pix3D presents a scenario with exact shape matches in simpler scenes than ScanNet. Patch2CAD performs notably better than Mesh R-CNN [19], and is competitive with Mask2CAD [30], whose full object matching approach well-suits the scenario with exact 3D matches.

**How does a patch-wise embedding mold the space for top-k retrieval?** We evaluate our patch-based image-CAD embedding space by analyzing the top-$k$ nearest neighbor CAD models retrieved for a given detected object on the ScanNet validation set. We visualize the top-8 retrieved CAD models for various image object detections in Figure 4. In contrast to the full-shape mapping of image-CAD established by Mask2CAD, our patch-wise embedding construction encourages more similarly structured CAD shapes to be voted for by the patches, resulting in geometrically consistent top-$k$.

Quantitatively, we analyze our top-$k$ shape retrieval by evaluating the recall from the $k$ retrieved shapes. We compare with the state-of-the-art Mask2CAD [30] approach in Figure 5, using $k = 1$ to 24. Our patch-based approach consistently produces more accurate shape retrieval.

**Limitations.** While our Patch2CAD approach demonstrates a more robust joint embedding space construction between images and CAD models, there are various directions for development. For instance, our patch-based retrieval can produce more robust CAD retrieval results, but cannot represent shapes that differ significantly from the database; we believe a part-based synthesis or deformation approach from our various patch retrievals holds promise. Additionally, our approach tackles object shape and structure, but does not represent the full scene geometry, which is an important direction towards comprehensive 3D perception.

**5. Conclusion**

In this paper, we present Patch2CAD, which establishes patch-based correspondence between 2D images and 3D CAD models for a robust construction of a shared embedding space to map between the two domains. This enables CAD-based understanding of the shapes of objects seen from a 2D image, representing each object as a posed, lightweight, complete mesh. We demonstrate that our patch-wise embedding learning can construct a more meaningful embedding space for nearest neighbor retrieval, and more robust shape estimation for complex real-world imagery under many occlusions. We believe that this brings understanding forward in bridging these domains of 2D-3D as well as real-synthetic, which opens avenues in domain transfer, content creation, and 3D scene understanding.

**Acknowledgements**

We would like to thank our colleagues at Google Research for their advice, and the support of the Bavarian State Ministry of Science and the Arts as coordinated by the Bavarian Research Institute for Digital Transformation (bidt) for Angela Dai.

---

**Table 4:** Mean F-score and category F-score on ScanNet [11] using groundtruth 2D detections and evaluating shape only.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total3D [42]</td>
<td>52.4</td>
<td>58.8</td>
<td>69</td>
</tr>
<tr>
<td>Mask2CAD [30]</td>
<td>60.6</td>
<td>63.1</td>
<td>64.4</td>
</tr>
<tr>
<td>Patch2CAD (Ours)</td>
<td>63.8</td>
<td>64.3</td>
<td>62.0</td>
</tr>
</tbody>
</table>

**Table 5:** Performance on Pix3D [52]. We report mean APmesh following [19, 30].

<table>
<thead>
<tr>
<th>Dataset</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mesh R-CNN [19]</td>
<td>17.2</td>
<td>51.2</td>
<td>7.4</td>
</tr>
<tr>
<td>Mask2CAD [30]</td>
<td>33.2</td>
<td>54.9</td>
<td>30.8</td>
</tr>
<tr>
<td>Patch2CAD</td>
<td>30.9</td>
<td>51.7</td>
<td>28.2</td>
</tr>
</tbody>
</table>

**Figure 5:** Shape retrieval comparison with Mask2CAD [30]
References


