3D Pose Estimation and 3D Model Retrieval for Objects in the Wild

Alexander Grabner[1] Peter M. Roth[1] Vincent Lepetit[2,1]

[1]Institute of Computer Graphics and Vision, Graz University of Technology, Austria

[2]Laboratoire Bordelais de Recherche en Informatique, University of Bordeaux, France

{alexander.grabner,pmroth,lepetit}@icg.tugraz.at

Abstract

We propose a scalable, efficient and accurate approach to retrieve 3D models for objects in the wild. Our contribution is twofold. We first present a 3D pose estimation approach for object categories which significantly outperforms the state-of-the-art on Pascal3D+. Second, we use the estimated pose as a prior to retrieve 3D models which accurately represent the geometry of objects in RGB images. For this purpose, we render depth images from 3D models under our predicted pose and match learned image descriptors of RGB images against those of rendered depth images using a CNN-based multi-view metric learning approach. In this way, we are the first to report quantitative results for 3D model retrieval on Pascal3D+, where our method chooses the same models as human annotators for 50% of the validation images on average. In addition, we show that our method, which was trained purely on Pascal3D+, retrieves rich and accurate 3D models from ShapeNet given RGB images of objects in the wild.

Figure 1: Given an RGB image (top), we predict a 3D pose and a 3D model for objects of different categories (bottom).

1. Introduction

Retrieving 3D models for objects in 2D images, as shown in Fig. 1, is extremely useful for 3D scene understanding, augmented reality applications and tasks like object grasping or object tracking. Recently, the emergence of large databases of 3D models such as ShapeNet [3] initiated substantial interest in this topic and motivated research for matching 2D images of objects against 3D models. However, there is no straight forward approach to compare 2D images and 3D models, since they have considerably different representations and characteristics.

One approach to address this problem is to project 3D models onto 2D images, which is known as rendering [24]. This converts the task to comparing 2D images, which is, however, still challenging, because the appearance of objects in real images and synthetic renderings can significantly differ. In general, the geometry and texture of available 3D models do not exactly match those of objects in real images. Therefore, recent approaches [2, 10, 23, 28] use convolutional neural networks (CNNs) [7, 8, 22] to extract features from images which are partly invariant to these variations. In particular, these methods compute image descriptors from real RGB images and synthetic RGB images which are generated by rendering 3D models under multiple poses. While this allows them to train a single CNN purely on synthetic data, there are two main disadvantages:

First, there is a significant domain gap between real and synthetic RGB images: Real images are affected by complex lighting, uncontrolled degradation and natural backgrounds. This makes it is hard to render photo-realistic images from the available 3D models. Therefore, using a single CNN for feature extraction from both domains is limited in performance, and even domain adaption [13] does not fully account for the different characteristics of real and synthetic images.

Second, processing renderings from multiple poses is computationally expensive. However, this step is mandatory, because the appearance of an object can significantly vary with the pose, and mapping images from all poses to a common descriptor does not scale to many categories [11].

To overcome these limitations, we propose to first predict the object pose and to then use this pose as an effective prior for 3D model retrieval. Inspired by recent works on instance pose estimation [4, 19], we present a robust 3D pose estimation approach for object categories based on virtual control points. More specifically, we use a CNN to predict the 2D projections of virtual 3D control points from which we recover the pose using a PnP algorithm. This approach does not only outperform the state-of-the-art for viewpoint estimation on Pascal3D+ [29], but also supports category-agnostic predictions. Having an estimate of the 3D pose makes our approach scalable, as it reduces the matching process to a single rendering per 3D model.

Additionally, we propose to render depth images instead of RGB images and to use different CNNs for feature extraction from the real and synthetic domain. Thus, we are not only able to deal with untextured models, but also to alleviate the domain gap. We implement our 3D model retrieval method using a multi-view metric learning approach, which is trained on real and synthetic data from Pascal3D+. In this way, we are the first to present quantitative results for 3D model retrieval on Pascal3D+. Moreover, we demonstrate that our approach retrieves rich and accurate 3D models from ShapeNet given unseen images from Pascal3D+. To summarize, we make the following contributions:

– We present a 3D pose estimation approach for object categories which significantly outperforms the state-of-the-art on Pascal3D+. Our method predicts virtual control points which generalize across categories making the approach scalable.

– We introduce a 3D model retrieval approach which utilizes a pose prior. For this purpose, we match learned image descriptors of RGB images against those of depth images rendered from 3D models under our predicted pose. In this way, we retrieve 3D models from ShapeNet which accurately represent the geometry of objects in RGB images, as shown in Fig. 1.

2. Related Work

Since there is a vast amount of literature on both 3D pose estimation and 3D model retrieval, we focus our discussion on recent works which target these tasks for object categories in particular.

2.1. 3D Pose Estimation

Many recent works only perform 3-DoF viewpoint estimation and predict the object rotation using regression, classification or hybrid variants of the two. [28] directly regresses azimuth, elevation and in-plane rotation using a CNN. [12] compares different variants and presents a regression approach which parameterizes each angle using trigonometric functions. [25, 26] perform viewpoint classification by discretizing the range of each angle into a number of disjoint bins and predicting the most likely bin using a CNN. [24] uses a fine-grained geometric structure aware classification, which encourages the correlation between bins of nearby views. [15] formulates the task as a hybrid classification/regression problem: In addition to viewpoint classification, a residual rotation is regressed for each angular bin, and the 3D dimensions of the object are predicted. [14] uses a slightly different parameterization and predicts a 2D translation to refine the object localization in a coarse-to-fine hybrid approach.

However, predicting a full 6-DoF pose instead of a 3-DoF viewpoint is desirable for many applications. Therefore, numerous methods compute both rotation and translation from 2D/3D keypoint correspondences. [18] recovers the pose from keypoint predictions and CAD models using a PnP algorithm. [26] presents a keypoint prediction approach that combines local keypoint estimates with a global viewpoint estimate. [17] predicts semantic keypoints and trains a deformable shape model which takes keypoint uncertainties into account.

These approaches rely on category-specific keypoints which do not generalize across categories. In the context of 3D pose estimation for object instances, [4] therefore considers virtual control points and predicts their 2D projections to estimate the pose from object parts. [19] takes a similar approach, but uses the corners of the object's 3D bounding box as virtual control points. This work inspired our approach, however, it is not directly applicable for object category pose estimation, since the ground truth 3D model of an object must be known at runtime.

2.2. 3D Model Retrieval

One intuitive approach to 3D model retrieval is to rely on classification. [14] performs fine-grained category recognition and provides a model for each category. [1] uses a linear classifier on mid-level representations of real images and renderings from multiple viewpoints to predict both shape and viewpoint.

However, retrieval via classification does not scale. Therefore, many recent methods take a metric learning approach. The most common strategy is to train a single CNN to extract features from real RGB images and RGB renderings. [2] uses a CNN pre-trained on ImageNet [21] as a feature extractor and matches features of real images against those of 3D models rendered under multiple viewpoints to predict both shape and viewpoint. [10] takes a similar approach, but uses a different network architecture for feature

extraction. [13] also employs a pre-trained CNN, but additionally performs non-linear feature adaption to overcome the domain gap between real and rendered images.

[28] finetunes a pre-trained CNN using lifted structure embedding [16] and averages the distance of a real image to renderings from multiple viewpoints to be more invariant to object pose. [23] presents a CNN architecture that combines information of renderings from multiple viewpoints into a single object pose invariant descriptor. [11] explicitly constructs an embedding space using a 3D similarity measure evaluated on clean 3D models and trains a CNN to map renderings with arbitrary backgrounds to the corresponding points in the embedding space.

While it is convenient to use RGB images, it is unclear how to deal with untextured 3D models or how to set the scene lighting. Therefore, other methods perform 3D model retrieval using depth instead of RGB images. [5] uses an ensemble of autoencoders followed by a domain adaption layer to match real depth images against depth images of 3D models. [31] computes image descriptors by fusing global autoencoder and local SIFT features of depth images. However, real depth images are not available in many scenarios.

Another approach which alleviates the domain gap and maps different representations to a common space is multi-view learning. [6] trains two different networks to map 3D voxel grids and RGB images to a low dimensional embedding space, where 3D model retrieval is performed by matching embeddings of real RGB images against those of voxel grids. [30] also presents a multi-view approach using two networks, but maps LD-SIFT features extracted from 3D models and depth images to a common space. In contrast to these methods, we map real RGB images and rendered depth images to a common representation. In this way, we do not need to perform computationally expensive 3D convolutions for high-resolution voxel grids and do not rely on real depth images.

3. 3D Pose Estimation and 3D Model Retrieval

Given an RGB image containing one or more objects, we want to retrieve 3D models with a geometry that corresponds well to the actual objects. Fig. 2 shows our proposed pipeline. We first estimate the 3D pose of an object from an image window roughly centered on the object. In this work, we assume the input image windows are known as in [29] or given by a 2D object detector [20]. Similar to previous works [15, 17, 26], we also assume the object category to be known, as it is a useful prior for both pose estimation and model retrieval. However, we also show that this information is not necessarily required in our approach. In fact, we can retrieve an accurate pose with only a marginal loss of accuracy, when the category is unknown.

After we estimated the object pose, we render a number of candidate 3D models under that pose. In particular, we render depth images, which allows us to deal with untextured 3D models and to circumvent the problem of scene lighting. In order to compare the real RGB image to synthetic depth renderings, we extract image descriptors using two CNNs, one for each domain. Finally, we match these image descriptors to retrieve the closest 3D model.

3.1. 3D Pose Estimation

The first step in our model retrieval approach is to robustly compute the 3D pose of the objects of interest. For this purpose, inspired by [4, 19], we predict the 2D image locations of virtual control points. More precisely, we train a CNN to predict the 2D image locations of the projections of the object's eight 3D bounding box corners. The actual 3D pose is then computed by solving a perspective-n-point (PnP) problem, which recovers rotation and translation from 2D-3D correspondences. This is illustrated in the first row of Fig. 2.

However, PnP algorithms require the 3D coordinates of the virtual control points to be known. Therefore, previous approaches either assume the exact 3D model to be given at runtime [19] or predict the projections of static 3D points [4]. To overcome this limitation, we predict the spatial dimensions $D = [d_x, d_y, d_z]$ of the object's 3D bounding box and use these to scale a unit cube, which approximates the ground truth 3D coordinates.

For this purpose, we introduce a CNN architecture which jointly predicts the 2D image locations of the projections of the eight 3D bounding box corners (16 values) as well as the 3D bounding box dimensions (3 values). As illustrated in Fig. 3, we implement this architecture as a single 19 neuron linear output layer, which we apply on top of the penultimate layer of different base networks such as VGG [22] or ResNet [7, 8]. During training, we optimize the pose loss

$$L_{\text{pose}} = L_{\text{proj}} + \alpha L_{\text{dim}} + \beta L_{\text{reg}} \,, \tag{1}$$

which is a linear combination of the projection loss L_{proj}, the dimension loss L_{dim} and the regularization L_{reg}. The meta-parameters α and β control the impact of the different loss terms. Let M_i be the i-th 3D bounding box corner and $\text{Proj}_{\text{R,t}}(\mathrm{M}_i)$ its projection using the ground truth rotation R and translation t, then the projection loss

$$L_{\text{proj}} = E\left[\sum_{i=1}^{8} \|\text{Proj}_{\text{R,t}}(\mathrm{M}_i) - \widetilde{m}_i\|_{\text{Huber}}\right] \tag{2}$$

is the expected value of the distances between the ground truth projections $\text{Proj}_{\text{R,t}}(\mathrm{M}_i)$ and the predicted locations of these projections $\widetilde{m}_i$ computed by the CNN for the training set. Being aware of inaccurate annotations in datasets such as Pascal3D+ [29], we use the Huber loss [9] in favor of the squared loss to be more robust to outliers.

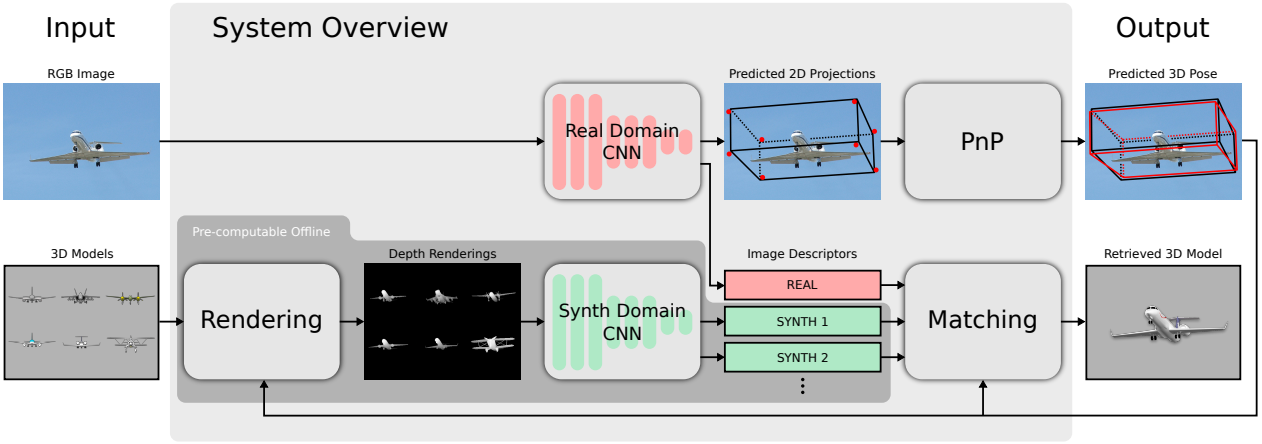

Figure 2: Overview of our approach. First row: Given an RGB image of an object, we first predict its 3D pose. We use a CNN to predict the 2D projections of the object's 3D bounding box corners (red dots). From these, we recover the object pose using a PnP algorithm. Second row: We render depth images from 3D models under the estimated pose and extract image descriptors from the real RGB image and the synthetic depth images using two different CNNs. Finally, we match the computed descriptors to retrieve the closest 3D model. Our approach supports pre-computed synthetic descriptors.

The dimension loss

$$L_{\text{dim}} = E\left[\sum_{i=x,y,z} \|d_i - \widetilde{d}_i\|_{\text{Huber}}\right] \tag{3}$$

is the expected value of the distances between the ground truth 3D dimensions d_i and the 3D dimensions $\widetilde{d}_i$ predicted by the CNN for the training set. To reduce the risk of overfitting, the regularization L_{reg} in Eq. (1) adds weight decay for all CNN parameters.

3.2. 3D Model Retrieval

Having a robust estimate of the object pose, we render 3D models under this pose instead of rendering them under multiple poses [2, 10, 13, 23]. This significantly reduces the computational complexity compared to methods which process multiple renderings for each 3D model and provides a useful prior for retrieval. In contrast to recent approaches [11, 13, 23, 28], we render depth images instead of RGB images. This allows us to deal with 3D models which do not have material or texture. Additionally, we circumvent the problem of how to set the scene lighting.

Before rendering a 3D model, we re-scale it to tightly fit into our predicted 3D bounding box. This is done by multiplying all vertices with the minimum of the ratio between the predicted 3D dimensions computed during pose estimation and the model's actual 3D dimensions. In this way, we improve the alignment between input RGB images and rendered depth images.

However, since RGB images and depth images have considerably different characteristics, we introduce a multi-view metric learning approach, which maps images from both domains to a common representation. We implement this mapping using a separate CNN for each domain. For real RGB images, we extract image descriptors from the hidden feature activations of the penultimate layer of our pose estimation CNN (see Fig. 3). As these activations have already been computed during pose estimation inference, we get the real image descriptor without any additional computational cost. For the synthetic depth images, we extract image descriptors using a CNN with the same architecture as our pose estimation CNN, except for the output layer (see Fig. 3).

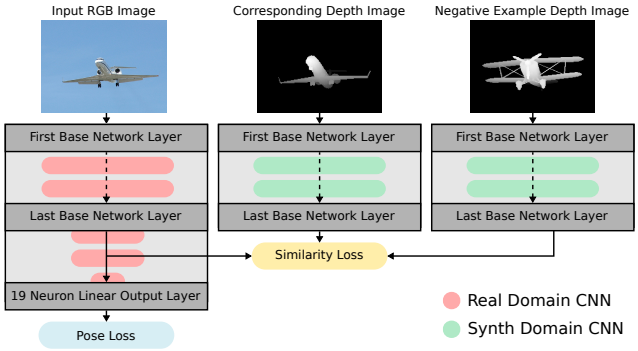

Figure 3: The pose loss is computed on the output of the real domain CNN. The similarity loss is computed on hidden feature maps extracted from the last base network layer of the real and synthetic domain CNN.

To finally map images from both domains to a common representation, we optimize the similarity loss

$$L_{\text{similarity}} = L_{\text{descr}} + \gamma L_{\text{reg}_2}\,, \tag{4}$$

which comprises the image descriptor loss L_{descr} and the regularization L_{reg_2} weighted by the meta-parameter γ.

The image descriptor loss

$$L_{\text{descr}} = E\left[\max(0, s^+ - s^- + m)\right] \quad (5)$$

minimizes the expected value of the Triplet loss [27] for the training set. Here, s^+ is the Euclidean distance between the real RGB image descriptor and the corresponding synthetic depth image descriptor, s^- is the Euclidean distance between the real RGB image descriptor and a negative example synthetic depth image descriptor, and m specifies the margin, *i.e.*, the desired minimum difference between s^+ and s^-. To reduce the risk of overfitting, the regularization L_{reg_2} in Eq. (4) adds weight decay for all CNN parameters.

After the optimization of the CNNs, we can pre-compute descriptors for synthetic depth images. In this case, we generate multiple renderings for each 3D model, which cover the full pose space. We then compute descriptors for all these renderings and store them in a database. At runtime, we just match descriptors from the viewpoint closest to our predicted pose, which is fast and scalable, but still accurate as shown in our experiments.

4. Experimental Results

To demonstrate our 3D model retrieval approach for objects in the wild, we evaluate it in a realistic setup where we retrieve 3D models from ShapeNet [3] given unseen RGB images from Pascal3D+ [29]. In particular, we train our 3D model retrieval approach purely on data from Pascal3D+, but use it to retrieve 3D models from ShapeNet. The corresponding results are detailed in Sec. 4.2. As estimating an accurate object pose is essential for our retrieval approach, we additionally evaluate our pose estimation approach on Pascal3D+ in Sec. 4.1.

4.1. 3D Pose Estimation

In the following, we first give a detailed evaluation of our pose estimation approach. Then, we compare it to previous methods, outperforming the state-of-the-art for viewpoint estimation on Pascal3D+. Finally, we demonstrate that we are even able to top the state-of-the-art without providing the correct category prior in some cases. For a fair evaluation, we follow the evaluation protocol of [26], which quantifies 3-DoF viewpoint prediction accuracy on Pascal3D+ using the geodesic distance

$$\Delta(R_{\text{gt}}, R_{\text{pred}}) = \frac{\|\log(R_{\text{gt}}^T R_{\text{pred}})\|_F}{\sqrt{2}} \quad (6)$$

to measure the difference between the ground truth viewpoint rotation matrix R_{gt} and the predicted viewpoint rotation matrix R_{pred}. In particular, we report two metrics: $MedErr$ (the median of all viewpoint differences) and $Acc_{\frac{\pi}{6}}$ (the percentage of all viewpoint differences smaller than $\frac{\pi}{6}$ respectively 30°). Evaluating our approach using the AVP metric [29], which couples 2D object detection and azimuth classification, is not meaningful as it is very different from our specific task.

	$MedErr$	$Acc_{\frac{\pi}{6}}$
Ours - VGG	11.7	0.8076
Ours - VGG+blur	11.6	0.8033
Ours - ResNet	**10.9**	0.8341
Ours - ResNet+blur	**10.9**	**0.8392**

Table 1: Viewpoint estimation using ground truth detections on Pascal3D+ for different setups of our approach. We report the mean performance across all categories.

4.1.1 3D Pose Estimation on Pascal3D+

Table 1 presents quantitative results for 3-DoF viewpoint estimation on Pascal3D+ using our approach in different setups, starting from a baseline using VGG to a more elaborated version building on ResNet. Specific implementation details and other parameters are provided in the supplementary material. For our baseline approach (*Ours - VGG*) we build on VGG and fine-tune the entire network for our task similar to [15, 25, 26]. As can be seen from Table 2, this baseline already matches the state-of-the-art.

When inspecting the failure cases, we see that many of them relate to small objects. In these cases, object image windows need to be upscaled to fit the fixed spatial input resolution of pre-trained CNNs. This results in blurry images and VGG, which only employs 3×3 convolutions, performs poorly at extracting features from over-smoothed images.

Therefore, we propose to use a network with larger kernel sizes that performs better at handling over-smoothed input images such as ResNet50 [7, 8], which uses 7×7 kernels in the first convolutional layer. As presented in Table 1, our approach with ResNet-backend (*Ours - ResNet*) significantly outperforms the VGG-based version. In addition, the total number of network parameters is notably lower (VGG: 135M vs. ResNet: 24M).

To further improve the performance, we employ data augmentation in the form of image blurring. Using ResNet as a base network together with blurring training images (*Ours ResNet+blur*), we improve on the $Acc_{\frac{\pi}{6}}$ metric while maintaining low $MedErr$ (see Table 1). This indicates that we improve the performance on over-smoothed images, but do not loose accuracy on sharp images. While our approach with ResNet-backend shows increased performance in this setup, we do not benefit from training on blurred images using a VGG-backend (*Ours - VGG+blur*). This also confirms that VGG is not suited for feature extraction from over-smoothed images. For all following experiments, we use our best performing setup, *i.e.*, *Ours - ResNet+blur*.

	category-specific												
	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
$MedErr$ ([17])	11.2	15.2	37.9	13.1	4.7	6.9	12.7	N/A	N/A	21.7	9.1	38.5	N/A
$MedErr$ ([17]*)	**8.0**	13.4	40.7	11.7	**2.0**	5.5	10.4	N/A	N/A	9.6	8.3	32.9	N/A
$MedErr$ ([26])	13.8	17.7	21.3	12.9	5.8	9.1	14.8	15.2	14.7	13.7	8.7	15.4	13.6
$MedErr$ ([15])	13.6	12.5	22.8	**8.3**	3.1	5.8	11.9	12.5	12.3	12.8	6.3	11.9	11.1
$MedErr$ ([24]**)	15.4	14.8	25.6	9.3	3.6	6.0	**9.7**	**10.8**	16.7	**9.5**	**6.1**	12.6	11.7
$MedErr$ (Ours)	10.0	15.6	**19.1**	8.6	3.3	**5.1**	13.7	11.8	**12.2**	13.5	6.7	**11.0**	**10.9**
$Acc_{\frac{\pi}{6}}$ ([26])	0.81	0.77	0.59	0.93	**0.98**	0.89	0.80	0.62	0.88	0.82	0.80	0.80	0.8075
$Acc_{\frac{\pi}{6}}$ ([15])	0.78	**0.83**	0.57	0.93	0.94	0.90	0.80	0.68	0.86	0.82	0.82	0.85	0.8103
$Acc_{\frac{\pi}{6}}$ ([24]**)	0.74	**0.83**	0.52	0.91	0.91	0.88	**0.86**	**0.73**	0.78	**0.90**	**0.86**	**0.92**	0.8200
$Acc_{\frac{\pi}{6}}$ (Ours)	**0.83**	0.82	**0.64**	**0.95**	0.97	**0.94**	0.80	0.71	0.88	0.87	0.80	0.86	**0.8392**
	category-agnostic												
	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
$MedErr$ (Ours)	10.9	**12.2**	23.4	9.3	3.4	5.2	15.9	16.2	**12.2**	11.6	6.3	11.2	11.5
$Acc_{\frac{\pi}{6}}$ (Ours)	0.80	0.82	0.57	0.90	0.97	**0.94**	0.72	0.67	**0.90**	0.80	0.82	0.85	0.8133

Table 2: Viewpoint estimation using ground truth detections on Pascal3D+. * The ground truth 3D model must be known at runtime. ** The approach was trained on vast amounts of RGB renderings from ShapeNet, instead of Pascal3D+ data.

4.1.2 Comparison to the State-of-the-Art

Next, we compare our pose estimation approach to state-of-the-art methods on Pascal3D+. Quantitative results are presented in Table 2. Our approach significantly outperforms the state-of-the-art in both $MedErr$ and $Acc_{\frac{\pi}{6}}$ considering mean performance across all categories and also shows competitive results for individual categories.

However, the $Acc_{\frac{\pi}{6}}$ scores for two categories, *boat* and *table*, are significantly below the mean. We analyze these results in more detail. The category *boat* is the most challenging category due to the large intra-class variability in shape and appearance. Many detections for this category are of low resolution and often objects are barely visible because of fog or mist. Additionally, there are a lot of ambiguities, *e.g.*, even a human cannot distinguish between the front and the back of an unmanned canoe. Nevertheless, we outperform the state-of-the-art for this challenging category.

The low $Acc_{\frac{\pi}{6}}$ scores for the category *table* can be explained by three factors. First, many tables are partly occluded by chairs (see *table* in Fig. 4). Second, the evaluation protocol does not take into account that many tables are ambiguous with respect to an azimuth rotation of π, $\frac{\pi}{2}$ or even have an axis of symmetry, *e.g.*, a round table. In some cases, our system predicts an ambiguous pose instead of the ground truth pose, while it is not possible to differentiate between the two poses. The evaluation protocol needs to be changed to take this into account. Last, the number of validation samples is very small (*i.e.*, 21) and, therefore, the reported results for this category are highly biased.

4.1.3 Category-Agnostic Pose Estimation

So far, the discussed results are category-specific, which means that the ground truth category must be known at runtime. In fact, all methods use a separate output layer for each category. However, our approach is able to make category-agnostic predictions which generalize across different categories. In this case, we use a single 19 neuron output layer which is shared for all categories making our approach scalable. Our category-agnostic pose estimation even outperforms the previous category-specific state-of-the-art for some categories, because it fully leverages the mutual information between similar categories like *bike* and *mbike*, for example, as shown in Table 2.

4.2. 3D Model Retrieval

Now we demonstrate our 3D model retrieval approach using our predicted pose. First, we present a quantitative evaluation of our approach on Pascal3D+. Second, we show qualitative results for 3D model retrieval from ShapeNet given images from Pascal3D+. Finally, we use our predicted 6-DoF pose and 3-DoF dimensions to precisely align retrieved 3D models with objects in real world images.

4.2.1 3D Model Retrieval from Pascal3D+

Since Pascal3D+ provides correspondences between RGB images and 3D models as well as pose annotations, we can train our approach purely on this dataset. In fact, we are the first to report quantitative results for 3D model retrieval

	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
Top-1-Acc (Rand)	0.15	0.21	0.36	0.25	0.25	0.10	0.15	0.10	0.28	0.31	0.27	0.27	0.2250
Top-1-Acc (Cano)	0.12	0.25	0.38	0.35	0.45	0.21	0.20	0.15	0.20	0.21	0.49	0.50	0.2925
Top-1-Acc (Off)	0.48	0.33	0.58	**0.41**	0.75	0.35	0.28	0.10	0.44	0.28	0.62	0.63	0.4375
Top-1-Acc (Pred)	0.48	0.31	**0.60**	**0.41**	0.78	0.41	0.29	0.19	0.43	**0.36**	0.65	0.61	0.4600
Top-1-Acc (GT)	**0.53**	**0.38**	0.51	0.37	**0.79**	**0.44**	**0.32**	**0.43**	**0.48**	0.33	**0.66**	**0.72**	**0.4967**

Table 3: 3D model retrieval accuracy using ground truth detections on Pascal3D+.

Figure 4: Qualitative results for 3D pose estimation and 3D model retrieval from ShapeNet given images from Pascal3D+ for all twelve categories. For each category, we show: the query RGB image; the depth image and RGB rendering of the ground truth 3D model from Pascal3D+ under the ground truth pose from Pascal3D+; the depth image and RGB rendering of our retrieved 3D model from ShapeNet under our predicted pose. We provide more results in the supplementary material.

on this dataset. For this purpose, we compute the top-1-accuracy (*Top-1-Acc*), *i.e.*, the percentage of evaluated samples for which the top retrieved model equals the ground truth model. This task is not trivial, because many models in Pascal3D+ have similar geometry and are hard to distinguish. Thus, we evaluate our approach using five different pose setups, *i.e.*, the ground truth pose (*GT*), our predicted pose (*Pred*), our predicted pose with offline pre-computed descriptors (*Off*), a canonical pose (*Cano*) and a random pose (*Rand*). Table 3 shows quantitative retrieval results.

As expected, we achieve the highest accuracy assuming the ground truth pose to be known (*GT*). In this case, our approach chooses the same 3D models as human annotators for 50% of the validation images on average. However, if we render the 3D models under our predicted pose (*Pred*), we almost match the accuracy of the ground truth pose setup. For some categories, we observe even better accuracy when using our predicted pose. This proves the high quality of our predicted pose. Moreover, our approach is fast and scalable at runtime while almost maintaining accuracy by using offline pre-computed descriptors (*Off*). For this experiment, we discretize the pose space in intervals of 10° and pre-compute descriptors for the 3D models. At runtime, we only match pre-computed descriptors from the discretized pose which is closest to our predicted pose and do not have to render 3D models online. If we, in contrast, just render the 3D models under a random pose (*Rand*) the performance decreases significantly. Rendering models under

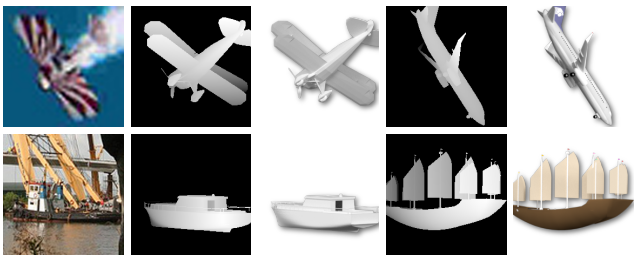

Figure 5: Example failure cases of our 3D model retrieval approach (same image arrangement as in Fig. 4). Top: The pose estimation fails because no similar pose was seen during training, as a result, the model retrieval fails. In this case, also the ground truth pose annotation from Pascal3D+ is not accurate. Bottom: While we estimate the pose correctly, the model retrieval fails due to heavy clutter.

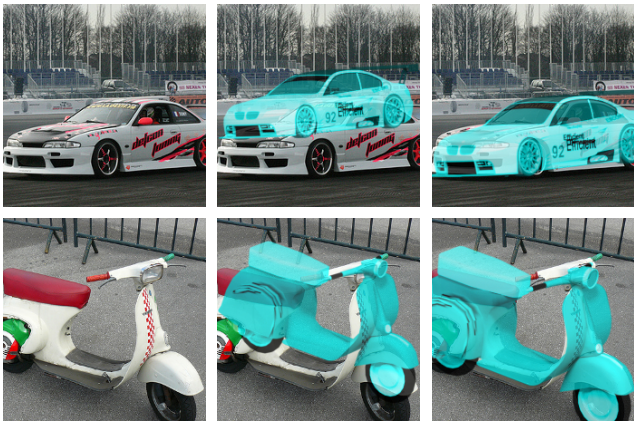

Figure 6: We use our predicted 6-DoF pose and 3-DoF dimensions to refine the alignment between the object and a rendering. Left: A detected object, which is not centered on the image window. Middle: A rendering which just uses our predicted 3-DoF rotation. Right: A rendering which uses our predicted 6-DoF pose and 3-DoF dimensions.

a frontal view (*Cano*) on the other hand provides a useful bias for the categories *train*, *bus* and *tv monitor* which are frequently seen from an almost frontal view in this dataset. These results confirm the importance of fine pose estimation in our approach.

4.2.2 3D Model Retrieval from ShapeNet

In contrast to Pascal3D+, ShapeNet provides a significantly larger spectrum of 3D models. Thus, we now evaluate our retrieval approach trained purely on Pascal3D+ for 3D model retrieval from ShapeNet given previously unseen images from Pascal3D+. Fig. 4 shows qualitative retrieval results for all twelve categories. Our approach predicts accurate 3D poses and 3D models for objects of different categories. In some cases, our predicted pose (see *sofa* in Fig. 4) or our retrieved model from ShapeNet (see *aeroplane* and *chair* in Fig. 4) are even more accurate than the annotated ground truth from Pascal3D+. While the geometry of the retrieved models corresponds well to the objects in the query images, the materials and textures typically do not. The reason for this is that we use depth images for retrieval, which do not include color information. This issue can be addressed by extracting texture information from the query RGB image or by performing retrieval with RGBD images. However, this is up to future research. Fig. 5 shows failure cases of our approach. If the pose estimation fails, the model retrieval becomes even more difficult. This is also reflected in Table 3, where we observe a strong decrease in performance when we render models without pose information (*Rand* and *Cano*). Also, if there is too much clutter in the query image, we cannot retrieve an accurate 3D model.

4.2.3 3D Model Alignment

Finally, we use our predicted 6-DoF pose and 3-DoF dimensions to precisely align retrieved 3D models with objects in real world images. Fig. 6 shows how we improve the 2D object localization and the alignment between the object and a rendering using our predicted pose and dimensions. This is especially useful if the object detection is not fully accurate, which is true in almost all situations. In this case, the detected image windows are a bit too small and the objects are not centered in the image windows. Thus, if we just render a model under our predicted rotation, re-scale it to tightly fit into the 2D image window, and center it in the 2D image window, the alignment is poor. However, if we additionally use our predicted translation and 3D dimensions for scaling and positioning, we significantly improve the alignment between the object and the rendering. This is of tremendous importance for robotics or augmented reality applications.

5. Conclusion

3D object retrieval from RGB images in the wild is an important but challenging task. Existing approaches address this problem by training on vast amounts of synthetic data. However, there is a significant domain gap between real and synthetic images which limits performance. For this reason, we learn to map real RGB images and synthetic depth images to a common representation. Additionally, we show that estimating the object pose is a useful prior for 3D model retrieval. Our approach is scalable as it supports category-agnostic predictions and offline pre-computed descriptors. We do not only outperform the state-of-the-art for viewpoint estimation on Pascal3D+, but also retrieve accurate 3D models from ShapeNet given unseen RGB images from Pascal3D+. Finally, these results motivate future research on jointly learning from real and synthetic data.

Acknowledgement This work was funded by the Christian Doppler Laboratory for Semantic 3D Computer Vision.

References

[1] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3D Chairs: Exemplar Part-Based 2D-3D Alignment Using a Large Dataset of Cad Models. In *Conference on Computer Vision and Pattern Recognition*, pages 3762–3769, 2014. 2

[2] M. Aubry and B. Russell. Understanding Deep Features with Computer-Generated Imagery. In *Conference on Computer Vision and Pattern Recognition*, pages 2875–2883, 2015. 1, 2, 4

[3] A. 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, Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 1, 5, 11

[4] A. Crivellaro, M. Rad, Y. Verdie, K. Moo Yi, P. Fua, and V. Lepetit. A Novel Representation of Parts for Accurate 3D Object Detection and Tracking in Monocular Images. In *International Conference on Computer Vision*, pages 4391–4399, 2015. 2, 3

[5] R. Girdhar, D. Fouhey, M. Rodriguez, and A. Gupta. Feng, Jie and Wang, Yan and Chang, Shih-Fu. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–9, 2016. 3

[6] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a Predictable and Generative Vector Representation for Objects. In *European Conference on Computer Vision*, pages 484–499, 2016. 3

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 3, 5, 13

[8] K. He, X. Zhang, S. Ren, and J. Sun. Identity Mappings in Deep Residual Networks. In *European Conference on Computer Vision*, pages 630–645, 2016. 1, 3, 5, 13

[9] P. Huber. Robust Estimation of a Location Parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964. 3

[10] H. Izadinia, Q. Shan, and S. Seitz. IM2CAD. In *Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2, 4

[11] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. Guibas. Joint Embeddings of Shapes and Images via CNN Image Purification. *ACM Transactions on Graphics*, 34(6):234, 2015. 2, 3, 4

[12] F. Massa, R. Marlet, and M. Aubry. Crafting a Multi-Task CNN for Viewpoint Estimation. In *British Machine Vision Conference*, pages 911–9112, 2016. 2

[13] F. Massa, B. Russell, and M. Aubry. Deep Exemplar 2D-3D Detection by Adapting from Real to Rendered Views. In *Conference on Computer Vision and Pattern Recognition*, pages 6024–6033, 2016. 1, 3, 4

[14] R. Mottaghi, Y. Xiang, and S. Savarese. A Coarse-To-Fine Model for 3D Pose Estimation and Sub-Category Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 418–426, 2015. 2

[15] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka. 3D Bounding Box Estimation Using Deep Learning and Geometry. In *Conference on Computer Vision and Pattern Recognition*, pages 7074–7082, 2017. 2, 3, 5, 6, 11

[16] H. Oh Song, Y. Xiang, S. Jegelka, and S. Savarese. Deep Metric Learning via Lifted Structured Feature Embedding. In *Conference on Computer Vision and Pattern Recognition*, pages 4004–4012, 2016. 3

[17] G. Pavlakos, X. Zhou, A. Chan, K. Derpanis, and K. Daniilidis. 6-DoF Object Pose from Semantic Keypoints. In *International Conference on Robotics and Automation*, pages 2011–2018, 2017. 2, 3, 6, 11

[18] B. Pepik, M. Stark, P. Gehler, T. Ritschel, and B. Schiele. 3D Object Class Detection in the Wild. In *Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–10, 2015. 2

[19] M. Rad and V. Lepetit. BB8: A Scalable, Accurate, Robust to Partial Occlusion Method for Predicting the 3D Poses of Challenging Objects Without Using Depth. In *International Conference on Computer Vision*, pages 3828–3836, 2017. 2, 3

[20] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 3

[21] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2

[22] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*, 2014. 1, 3, 13

[23] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In *Conference on Computer Vision and Pattern Recognition*, pages 945–953, 2015. 1, 3, 4

[24] H. Su, C. Qi, Y. Li, and L. Guibas. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views. In *International Conference on Computer Vision*, pages 2686–2694, 2015. 1, 2, 6, 11

[25] S. Tulsiani, J. Carreira, and J. Malik. Pose Induction for Novel Object Categories. In *International Conference on Computer Vision*, pages 64–72, 2015. 2, 5

[26] S. Tulsiani and J. Malik. Viewpoints and Keypoints. In *Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015. 2, 3, 5, 6, 11

[27] K. Weinberger and L. Saul. Distance Metric Learning for Large Margin Nearest Neighbor Classification. *Journal of Machine Learning Research*, 10:207–244, 2009. 5

[28] 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, 2016. 1, 2, 3, 4

[29] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond Pascal: A Benchmark for 3D Object Detection in the Wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014. 2, 3, 5, 11

[30] J. Zhu, F. Zhu, E. Wong, and Y. Fang. Learning Pairwise Neural Network Encoder for Depth Image-Based 3D Model Retrieval. In *ACM International Conference on Multimedia*, pages 1227–1230, 2015. 3

[31] J. Zhu, F. Zhu, E. Wong, and Y. Fang. Deep Learning Representation Using Autoencoder for 3D Shape Retrieval. *Neurocomputing*, 204:41–50, 2016. 3

3D Pose Estimation and 3D Model Retrieval for Objects in the Wild Supplementary Material

In the following, we provide additional qualitative results for our 3D model retrieval approach in Sec. 6, which complement those presented in the paper. Furthermore, we analyze failure cases for both 3D model retrieval and the underlying 3D pose estimation in Sec. 7. Finally, in Sec. 8 we discuss implementation details, parameter choices, and other relevant settings.

6. 3D Model Retrieval

Fig. 7 shows additional qualitative results for 3D model retrieval from ShapeNet [3] given previously unseen images from Pascal3D+ [29] validation data for all twelve categories. Our approach predicts accurate 3D poses and 3D models for objects of different categories.

Fig. 8 presents further 3D model alignment results for object detections which are not fully accurate. We significantly improve the alignment between the object in the image and an RGB rendering of our retrieved 3D model by taking advantage of our predicted 6-DoF pose and 3-DoF dimensions compared to just using a 3-DoF viewpoint.

7. Failure Modes

Most failure cases of our 3D pose estimation on Pascal3D+ relate to low-resolution or ambiguous objects.

Fig. 9 shows 3D pose estimation results on low-resolution image windows from Pascal3D+ validation data. After re-scaling, the over-smoothed input RGB images lack details and sharp discontinuities, which results in incorrect pose predictions. In fact, even for a human it is difficult to identify the correct object poses in these examples.

Fig. 10 shows additional failure cases, observing that heavy occlusions, bad illumination conditions and difficult object poses, which are far from the poses seen during training, result in incorrect pose predictions.

As shown in Fig. 11, some objects from Pascal3D+ are symmetrical, which makes their poses not well defined. For example, it is impossible to differentiate between the front and back of a symmetric unmanned boat. This issue is even more apparent for tables: Many tables are ambiguous with respect to an azimuth rotation of π, $\frac{\pi}{2}$ or even have an axis of symmetry, such as a round table. When our approach predicts one of the possible poses that is not the annotated ground truth pose, this is considered as a mistake by the commonly used evaluation protocol [26].

Fig. 12 shows that visual distortions due to wide-angle lenses (*i.e.*, fish-eye effects), deformed and demolished objects and heavy occlusions can disturb the model retrieval step, even if the pose estimation was successful.

8. Implementation Details

In the following, we provide implementation details and other parameters used in our work:

Intrinsic camera parameters: In Pascal3D+, the ground truth poses were computed from 2D-3D correspondences assuming the same intrinsic parameters for all images. We employ the same parameters in our approach.

Data augmentation: Like others [15, 17, 24, 26], we perform data augmentation by jittering ground truth detections and exclude detections marked as occluded or truncated from the evaluation. Additionally, we augment samples for which the longer edge of the ground truth image window is greater than 224 pixel by applying Gaussian blurring with various kernel sizes and σ. We randomly sample negative example 3D models from the available data. All augmentation parameters are randomized after each training epoch.

Meta parameters: We normalize the projections so that the image pixel range is mapped to the interval [0,1] and use the same Huber loss ($\delta = 0.01$) for all 19 estimated values. Experimentally, we found $\alpha = 1$, $\beta = 1e^{-5}$ and $\gamma = 1e^{-3}$ to work well and set $m = 1$.

Network parameters: We use a batch size of 50, train our networks for 100 epochs and decrease the initial learning rate of $1e^{-4}$ by one order of magnitude after 50 and 90 epochs, and employ the Adam optimization algorithm.

3D dimensions: For both Pascal3D+ and ShapeNet, 3D models are normalized to fit within a unit cube centered at the origin. Thus, we estimate 3D dimensions in model space in the range [0,1]. Since these dimensions tend to be consistent within a category, estimating them is not a major issue. Table 4 shows quantitative results for 3D dimension estimation. We achieve high accuracy across all categories.

	x	y	z
Median Absolute Error	0.022	0.015	0.014

Table 4: 3D dimension estimation errors on Pascal3D+. We report the mean performance across all categories.

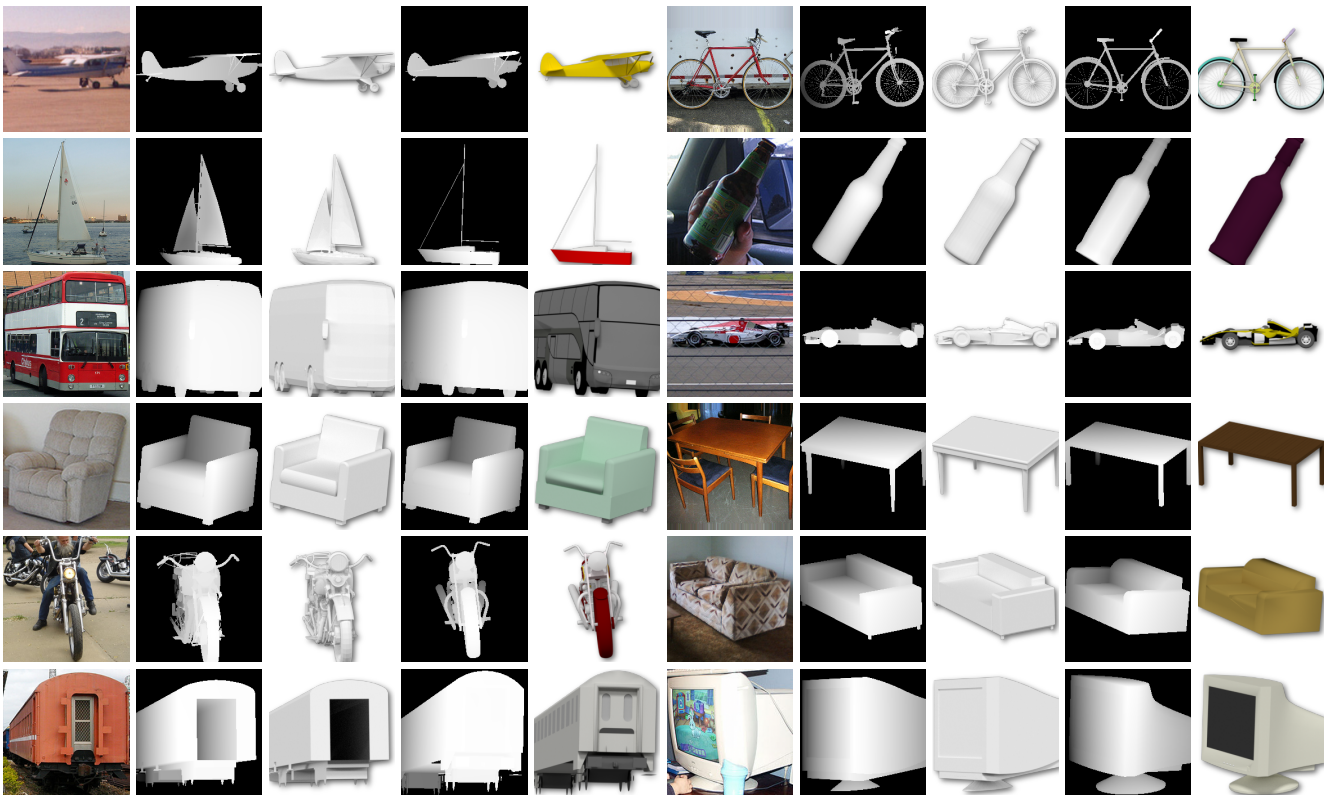

Figure 7: Qualitative results for 3D pose estimation and 3D model retrieval from ShapeNet given images from Pascal3D+ for all twelve categories. For each category, we show: the query RGB image; the depth image and RGB rendering of the ground truth 3D model from Pascal3D+ under the ground truth pose from Pascal3D+; the depth image and RGB rendering of our retrieved 3D model from ShapeNet under our predicted pose.

Figure 8: We use our predicted 6-DoF pose and 3-DoF dimensions to refine the alignment between the object and a rendering. Left: A detected object, which is not centered on the image window. Middle: A rendering which just uses our predicted 3-DoF rotation. Right: A rendering which uses our predicted 6-DoF pose and 3-DoF dimensions.

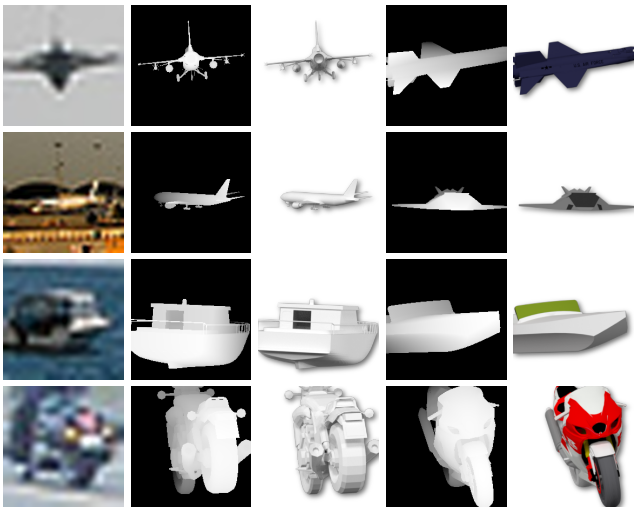

Figure 9: 3D pose estimation fails due to low-resolution image windows (same image arrangement as in Fig. 7). In fact, for more than 55% of Pascal3D+ validation detections the longer edge of the 2D image window is smaller than 224 pixel, which is the fixed spatial input size of pre-trained CNNs like VGG [22] or ResNet [7, 8]. If the resolution is too low, we cannot predict an accurate 3D pose.

Figure 10: 3D pose estimation fails in difficult situations (same image arrangement as in Fig. 7). We observe that heavy occlusions (first row), bad illumination conditions (second row) and difficult object poses (third and fourth row), which are far from the poses seen during training, result in incorrect pose predictions. In the last row, we see that not even the annotated ground truth pose is correct.

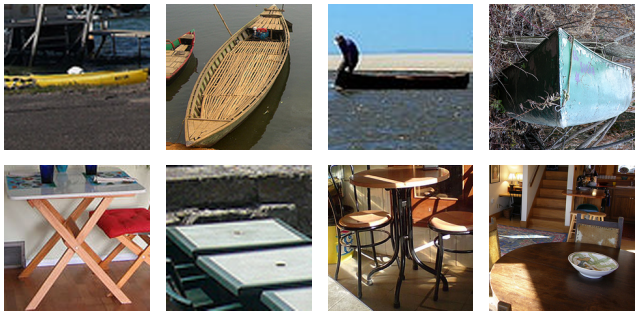

Figure 11: Objects with ambiguous poses from Pascal3D+ validation data. First row: It is impossible to differentiate between the front and back of symmetric boats. Second row: Tables which are ambiguous with respect to an azimuth rotation of π (first image), $\frac{\pi}{2}$ (second and third image) or even have an axis of symmetry (fourth image).

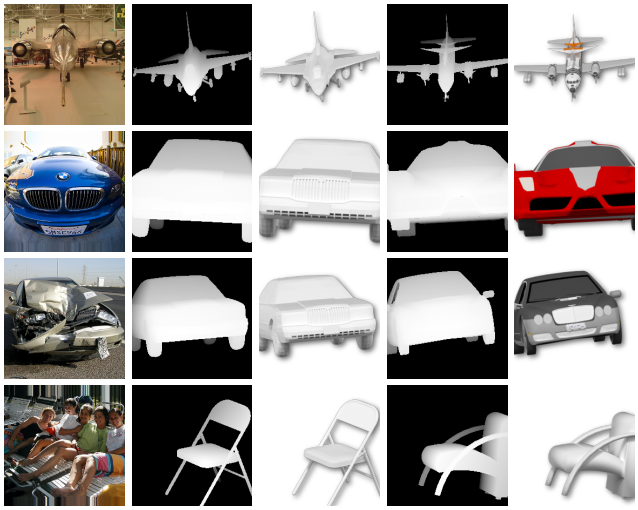

Figure 12: 3D model retrieval results for challenging cases where pose estimation was successful (same image arrangement as in Fig. 7). The test images can exhibit fish-eye effects due to wide-angle lenses (first and second row), contain deformed or demolished objects (third row), or objects under heavy occlusions (fourth row), which disturb object retrieval. Note however that the ground truth 3D models are not accurate.