Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views

Hao Su*, Charles R. Qi*, Yangyan Li, Leonidas J. Guibas
Stanford University

Abstract

Object viewpoint estimation from 2D images is an essential task in computer vision. However, two issues hinder its progress: scarcity of training data with viewpoint annotations, and a lack of powerful features. Inspired by the growing availability of 3D models, we propose a framework to address both issues by combining render-based image synthesis and CNNs (Convolutional Neural Networks). We believe that 3D models have the potential in generating a large number of images of high variation, which can be well exploited by deep CNN with a high learning capacity. Towards this goal, we propose a scalable and overfit-resistant image synthesis pipeline, together with a novel CNN specifically tailored for the viewpoint estimation task. Experimentally, we show that the viewpoint estimation from our pipeline can significantly outperform state-of-the-art methods on PASCAL 3D+ benchmark.

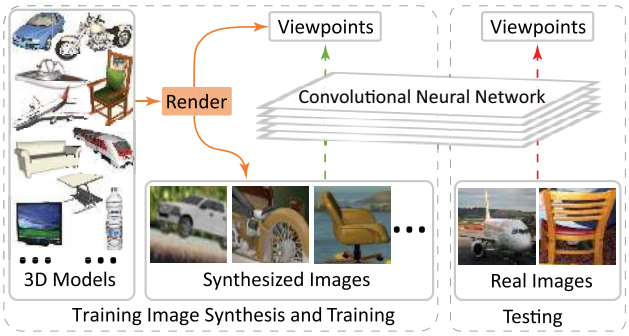

Figure 1. **System overview.** We synthesize training images by overlaying images rendered from large 3D model collections on top of real images. CNN is trained to map images to the ground truth object viewpoints. The training data is a combination of real images and synthesized images. The learned CNN is applied to estimate the viewpoints of objects in real images. Project homepage is at https://shapenet.cs.stanford.edu/projects/RenderForCNN

1. Introduction

3D recognition is a cornerstone problem in many vision applications and has been widely studied. Despite its critical importance, existing approaches are far from robust when applied to cluttered real-world images. We believe that two issues have to be addressed to enable more successful methods: scarcity of training images with accurate viewpoint annotation, and a lack of powerful features specifically tailored for 3D tasks.

The first issue, scarcity of images with accurate viewpoint annotation, is mostly due to the high cost of manual annotation, and the associated inaccuracies due to human error. Consequently, the largest 3D image dataset, PASCAL 3D+ [39], contains only ∼22K images. As such, it is limited in diversity and scale compared with object classification datasets such as ImageNet, which contains millions of images [5].

The second issue is a lack of powerful features specifically tailored for viewpoint estimation. Most 3D vision systems rely on features such as SIFT and HoG, which were designed primarily for classification and detection tasks. However, this is contrary to the recent finding — features learned by task-specific supervision leads to much better task performance [16, 11, 14]. Ideally, we want to learn stronger features by deep CNN. This, however, requires huge amount of viewpoint-annotated images.

In this paper, we propose to address both issues by combining render-based image synthesis and CNNs, enabling us to learn discriminative features. We believe that 3D models have the potential to generate large number of images of high variation, which can be well exploited by deep CNN with a high learning capacity.

The inspiration comes from our key observation: more and more high-quality 3D CAD models are available online. In particular, many geometric properties, such as symmetry and joint alignment, can be efficiently and reliably estimated by algorithms with limited human effort (Sec 2). By rendering the 3D models, we convert the rich information carried by them into 3D annotations automatically.

To explore the idea of "Render for CNN" for 3D tasks, we focus on the viewpoint estimation problem — for an input RGB image and a bounding box from an off-the-shelf detector, our goal is to estimate the viewpoint.

To prepare training data for this task, we augment real

*Indicates equal contributions.

images by synthesizing *millions* of highly diverse images. Several techniques are applied to increase the diversity of the synthesized dataset, in order to prevent the deep CNN from picking up unreliable patterns and push it to learn more robust features.

To fully exploit this large-scale dataset, we design a deep CNN specifically tailored for the viewpoint estimation task. We formulate a class-dependent *fine-grained viewpoint* classification problem and solve the problem with a novel loss layer adapted for this task.

The results are surprising: *trained on millions of "rendered" images, our CNN-based viewpoint estimator significantly outperforms state-of-the-art methods, tested on "real" images* from the challenging PASCAL 3D+ dataset.

In summary, our contributions are as follows:

- We show that training CNN by massive synthetic data is an effective approach for 3D viewpoint estimation. In particular, we achieve state-of-the-art performance on benchmark data set;
- Based upon existing 3D model repositories, we propose a synthesis pipeline that generates millions of images with accurate viewpoint labels at negligible human cost. This pipeline is scalable, and the generated data is resistant to overfitting by CNN;
- Leveraging on the big synthesized data set, we propose a fine-grained view classification formulation, with a loss function encouraging strong correlation of nearby views. This formulation allows us to accurately predict views and capture underlying viewpoint ambiguities.

2. Related Work

3D Model Datasets Prior work has focused on manually collecting organized 3D model datasets (e.g., [7, 10]).

Recently, several large-scale online 3D model repositories have grown to tremendous sizes through public aggregation, including the Trimble 3D warehouse (above 2.5M models in total), Turbosquid (300K models) and Yobi3D (1M models). Using data from these repositories, [38] built a dataset of $\sim$130K models from over 600 categories. More recently, ShapeNet [33] annotated $\sim$330K models from over 4K categories. Using geometric analysis techniques, they semi-automatically aligned 57K models from 55 categories by orientation.

3D Object Detection Most 3D object detection methods are based on representing objects with discriminative features for points [4], patches [6] and parts [19, 29, 34], or by exploring topological structures [15, 2, 3]. More recently, 3D models have been used for supervised learning of appearance and geometric structure. For example, [32] and [20] proposed similar methods that learn a 3D deformable part model and demonstrate superior performance for cars and chairs, respectively; [21] and [1] formulated an alignment problem and built key point correspondences between 2D images and rendered 3D views. In contrast to these prior efforts that use hand-designed models based on hand-crafted features, we let CNNs learn viewpoint estimation directly from data.

Synthesizing Images for Training Synthetic images rendered from 3D models have long been used in computer vision [25, 26, 37, 23]. Recently, [32, 19, 13, 27] used 3D models to render images for training object detectors and viewpoint classifiers. They tweak the rendering parameters to maximize model usage, since they have a limited number of 3D models - typically below 50 models per category and insufficient to capture the geometric and appearance variance of objects in practice. Leveraging 3D repositories, [20, 1] use 250 and 1.3K chair models respectively to render tens of thousands training images, which are then used to train deformable part models (DPM [8]). In our work, we synthesize several orders of magnitude more images than existing work. We also explore methods to increase data variation by changing background patterns, illumination, viewpoint, etc., which are critical for preventing overfitting of the CNN.

While both [27] and our work connect synthetic images with CNN, they are fundamentally different in task, approach and result. First, [27] focused on *2D* object detection, whereas our work focuses on *3D* viewpoint estimation. Second, [27] used a small set of synthetic images (2,000 in total) to train *linear classifiers* based on features extracted by *out-of-the-box* CNNs [16, 11]. In contrast, we develop a scalable synthesis pipeline and generate more than two million images to learn geometric-aware features by training *deep CNNs* (initialized by [11]). Third, the performance of [27], though better than previous work *using synthetic data*, did not match R-CNN baseline trained by real images [11]. In contrast, we show significant performance gains (Sec 5.2) over previous work [39] *using full set of real data* of PASCAL VOC 2012 (trainset).

3. Problem Statement

For an input RGB image, our goal is to estimate its viewpoint. We parameterize the viewpoint as a tuple (θ, ϕ, ψ) of camera rotation parameters, where θ is azimuth (longitude) angle, ϕ is elevation (latitude) angle, and ψ is in-plane rotation (around optical axis) angle. They are discretized in a *fine-grained* manner, with azimuth, elevation and in-plane rotation angles being divided into N bins respectively. The viewpoint estimation problem is formalized as classifying the camera rotation parameters into these fine-grained bins (classes).

By adopting a fine-grained (N=360) viewpoint classification formulation, our estimation is informative and accurate. Compared with regression-based formulations [22], our formulation returns the probabilities of each viewpoint,

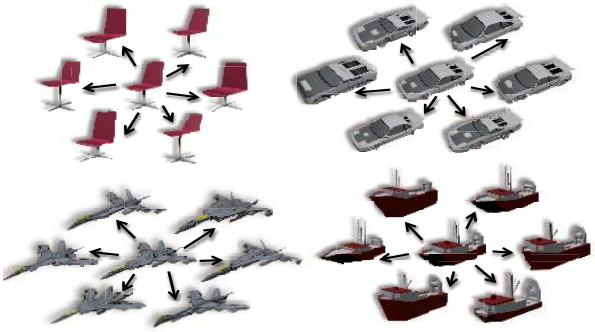

Figure 2. **3D model set augmentation by symmetry-preserving deformation.**

Figure 3. **Synthetic image examples.** Three example images are shown for each of the 12 classes from PASCAL 3D+.

thus capturing the underlying viewpoint ambiguity possibly caused by symmetry or occlusion patterns. This information can be useful for further processing. Compared with traditional coarse-grained classification-based formulations that typically have 8 to 24 discrete classes [29, 39], our formulation is capable of producing much more fine-grained viewpoint estimation.

4. Render for CNN System

Since the space of viewpoint is discretized in a highly fine-grained manner, massive training data is required for the training of the network. We describe how we synthesize such large number of training images in Sec 4.1, and how we design the network architecture and loss function for training the CNN with the synthesized images in Sec 4.2.

4.1. Training Image Generation

To generate training data, we render 3D model views. To increase the diversity of object geometry, we create new 3D models by deforming existing ones downloaded from a modestly-sized online 3D model repository. To increase the diversity of object appearance and background clutterness, we design a synthesis pipeline by randomly sampling rendering parameters, applying truncation patterns and adding random backgrounds from scene images.

Structure-preserving 3D Model Set Augmentation We take advantage of an online 3D model repository, ShapeNet, to collect seed models for classes of interest. The provided models are already aligned by orientation. For models that are bilateral or cylinder symmetric, their symmetry planes or axes are also already extracted. Please refer to [33] for more details of ShapeNet.

From each of the seed models, we generate new models by a structure-preserving deformation. The problem of structure-preservation deformation has been widely studied in the field of geometry processing, and there exists many candidate models as in survey [24]. We choose a symmetry-preserving free-form deformation, defined via regularly placed control points in the bounding cube, similar to the approach of [30]. Our choice is largely due to the model's simplicity and efficiency. More advanced methods can detect and preserve more structures, such as partial symmetry and rigidity [31].

To generate a deformed model from a seed model, we draw i.i.d samples from a Gaussian distribution for the translation vector of each control point. In addition, we regularize the deformation to set the translations of symmetric control points to be equal. Figure 2 shows example deformed models from our method.

Overfit-Resistant Image Synthesis We synthesize a large number of images for each 3D model and millions in total. *Rather than pursuing realistic effect, we try to generate images of high diversity to prevent deep CNNs from picking up unreliable patterns.*

We inject randomness in the three basic steps of our pipeline: rendering, background synthesis, and cropping.

For image rendering, we explore two sets of parameters, lighting condition and camera configuration. For the lighting condition, the number of light sources, their positions and energies are all sampled. For the camera extrinsics, we sample azimuth, elevation and in-plane rotation from distributions estimated from a real image training set. Refer to the supplementary material for more details.

Images rendered as above have a fully transparent background, and the object boundaries are highly contrasted. To prevent classifiers from overfitting such unrealistic boundary patterns, we synthesize the background by a simple and scalable approach. For each rendered image, we randomly sample an image from SUN397 dataset [40]. We use alpha-composition to blend a rendered image as foreground and a scene image as background.

To teach CNN to recognize occluded or truncated images, we crop the image by a perturbed object bounding box. The cropping parameters are also learned from a real image training set. We find that the cropped patterns tend

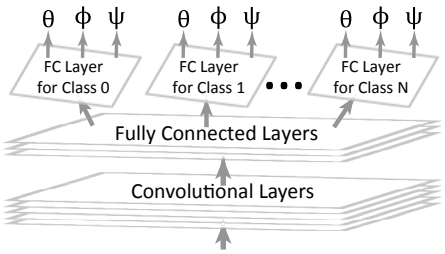

Figure 4. **Network architecture**[2]. Our network includes shared feature layers and class-dependent viewpoint estimation layers.

to be natural. For example, more bottom parts of chairs are cropped, since chair legs and seats are often occluded.

Finally, we put together the large amount of synthetic images, together with a small amount of real images with ground truth human annotations, to form our training image set. The ground truth annotation of a sample s is denoted as (c_s, v_s), where c_s is the class label of the sample, and $v_s \in \mathcal{V}$ is the the discretized viewpoint label tuple, and $\mathcal{V}$ is the space of discretized viewpoints.

4.2. Network Architecture and Loss Function

Class-Dependent Network Architecture. To effectively exploit our large-scale dataset, we need a model with sufficient learning capacity. CNN is the natural choice for this challenge. We adapt network structure of AlexNet [17] to fit our viewpoint estimation task.

We found that the CNN trained for viewpoint estimation of one class do not perform well on another class, possibly due to the huge geometric variation between the classes. Instead, the viewpoint estimation classifiers are trained in a class-dependent way. However, a naive way of training the class-dependent viewpoint classifiers, i.e., one network for each class, cannot scale up, as the parameter of the whole system increases linearly with the number of classes.

To address this issue, we propose a novel network architecture where lower layers (both convolutional layers and fully connected layers, i.e. conv1-conv5 and fc6-fc7) are shared by all classes, while class-dependent layers (one fc layer for each class k, $k = 0, .., N$) are stacked over them (see Figure 4). During training, loss from class k will only affect its corresponding top fc layer and the lower shared layers. Our network architecture design accommodates the fact that viewpoint estimation are class-dependent and maximizes the usage of low-level features shared across different classes to keep the overall network parameter number tractable. We initialize the shared convolutional and fully connected layers with the weights from [12]. During training, for real images all convolutional and fully connected layers are fine-tuned, for rendered images, last two conv layers and all fc layers are fine tuned. The class-dependent fully connected layers are trained from scratch.

Geometric Structure Aware Loss Function. The outputs of the network, the (θ, ϕ, ψ) tuples, are geometric entities. We propose a geometric structure aware loss function to exploit their geometric constraints. We define the viewpoint classification loss L_{vp} adapted from the soft-max loss as:

$$L_{vp}(\{s\}) = -\sum_{\{s\}} \sum_{v \in \mathcal{V}} e^{-d(v, v_s)/\sigma} \log P_v(s; c_s), \quad (1)$$

where $P_v(s; c_s)$ is the probability of view v for sample s from the soft-max viewpoint classifier of class c_s, and $d : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}$ is the distance between two viewpoints, e.g. geodesic distance on 3-sphere or geodesic distance of points by (θ, ϕ) on 2-sphere plus scaled ℓ_1 distance of ψ. By substituting an exponential decay weight w.r.t viewpoint distance for the mis-classification indicator weight in the original soft-max loss, we explicitly encourage correlation among the viewpoint predictions of nearby views.

5. Experiments

Our experiments are divided into four parts. First, we evaluate our viewpoint estimation system on the PASCAL3D+ data set [39] (Sec 5.2). Second, we visualize the structure of the learned viewpoint-discriminative feature space (Sec 5.3). Third, we perform control experiments to study the effects of synthesis parameters (Sec 5.4). Last, we show more qualitative results and analyze error patterns. (Sec 5.5). Before we discuss experiment details, we first overview the 3D model set used in all the experiments.

5.1. 3D Model Dataset

As we discussed in Sec 2, there are several large-scale 3D model repositories online. We download 3D models from ShapeNet [33], which has organized common daily objects with categorization labels and joint alignment. Since we evaluate our method on the PASCAL 3D+ benchmark, we download 3D models belonging to the 12 categories of PASCAL 3D+, including 30K models in total. After symmetry-preserving model set augmentation (Sec 4.1), we make sure that every category has 10K models. For more details, please refer to supplementary material.

5.2. Comparison with state-of-the-art Methods

We compare with state-of-the-art methods on PASCAL 3D+ benchmark.

Methods in Comparison We compare with two baseline methods, VDPM [39] and DPM-VOC+VP [28], trained on

[2]For simplicity, Pooling, Dropout, and ReLU layers are not shown. See the supplementary material for the full network definition.

VOC 2012 val AVP	aero	bicycle	boat	bus	car	chair	table	mbike	sofa	train	tv	Avg.
VDPM-4V [39]	34.6	41.7	1.5	26.1	20.2	6.8	3.1	30.4	5.1	10.7	34.7	19.5
VDPM-8V	23.4	36.5	1.0	35.5	23.5	5.8	3.6	25.1	12.5	10.9	27.4	18.7
VDPM-16V	15.4	18.4	0.5	46.9	18.1	6.0	2.2	16.1	10.0	22.1	16.3	15.6
VDPM-24V	8.0	14.3	0.3	39.2	13.7	4.4	3.6	10.1	8.2	20.0	11.2	12.1
DPM-VOC+VP-4V [9]	37.4	43.9	0.3	48.6	36.9	6.1	2.1	31.8	11.8	11.1	32.2	23.8
DPM-VOC+VP-8V	28.6	40.3	0.2	38.0	36.6	9.4	2.6	32.0	11.0	9.8	28.6	21.5
DPM-VOC+VP-16V	15.9	22.9	0.3	**49.0**	29.6	6.1	2.3	16.7	7.1	20.2	19.9	17.3
DPM-VOC+VP-24V	9.7	16.7	2.2	42.1	24.6	4.2	2.1	10.5	4.1	20.7	12.9	13.6
Ours-Joint-4V	**54.0**	**50.5**	**15.1**	**57.1**	**41.8**	**15.7**	**18.6**	**50.8**	**28.4**	**46.1**	**58.2**	**39.7**
Ours-Joint-8V	**44.5**	**41.1**	**10.1**	**48.0**	**36.6**	**13.7**	**15.1**	**39.9**	**26.8**	**39.1**	**46.5**	**32.9**
Ours-Joint-16V	**27.5**	**25.8**	**6.5**	45.8	**29.7**	**8.5**	**12.0**	**31.4**	**17.7**	**29.7**	**31.4**	**24.2**
Ours-Joint-24V	**21.5**	**22.0**	**4.1**	**38.6**	**25.5**	**7.4**	**11.0**	**24.4**	**15.0**	**28.0**	**19.8**	**19.8**

Table 1. **Simultaneous object detection and viewpoint estimation on PASCAL 3D+**. The measurement is AVP (an extension of AP, where true positive stands only when bounding box localization *AND* viewpoint estimation are both correct). We show AVPs for four quantization cases of 360-degree views (into 4, 8, 16, 24 bins respectively, with increasing difficulty). Our method uses joint real and rendered images and trains a CNN tailored for this task.

real images from PASCAL 3D+ VOC 2012 train set and tested on VOC 2012 val.

For our method, we train on a combination of real images (around 12K images in VOC12 train set) and synthetic images. We synthesized 200K images per category and in total 2.4M images for all 12 classes, all of which are with accurate and free viewpoint and category annotations.

Joint Detection and Viewpoint Estimation Following the protocol of [39, 28], we test on the joint detection and viewpoint estimation task. The bounding boxes of baseline methods are from their detectors and ours are from R-CNN with bounding box regression [11].

We use AVP (Average Viewpoint Precision) advocated by [39] as evaluation metric. AVP is the average precision with a modified true positive definition, requiring both 2D detection *AND* viewpoint estimation to be correct.

Table 1 and Figure 5 summarize the results. We observe that our method trained with a combination of real images and rendered images significantly outperform the baselines by a large margin, from a coarse viewpoint discretization (4V) to a fine one (24V), in all object categories.

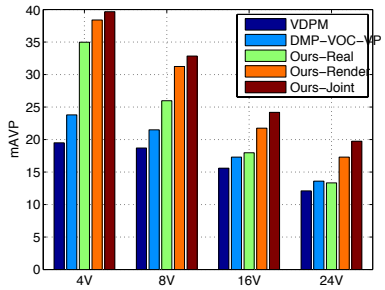

Figure 5. **Simultaneous object detection and viewpoint estimation performance.** We compare mAVP of our models and the state-of-the-art methods. We also compare Ours-Real with Ours-Render and Ours-Joint (use both real and rendered images for training) to see how much rendered images can help.

Viewpoint Estimation One might argue higher AVP is due to higher 2D detection performance of R-CNN. So we also directly compare viewpoint estimation performance using the same bounding boxes.

We first show the comparison results with VDPM, using the bounding boxes from R-CNN detection. For two sets from detection, only correctly detected bounding boxes are used (50% overlap threshold). The evaluation metric is a continuous version of viewpoint estimation accuracy, i.e., the percentage of bounding boxes whose prediction is within θ degrees of the ground truth.

Figure 6 summarizes the results. Again, our method is significantly better than VDPM on all sets. In particular, the median of the viewpoint estimation error for our method is 14°, which is much less than VDPM, being 57°.

Next we show the performance comparison using ground truth bounding boxes (Table 2). We compare with a recent work from [35], which uses a similar network architecture (TNet/AlexNet) as ours except the loss layer. Note that the viewpoint estimation in this experiment includes azimuth, elevation and in-plane rotation. We use the same metrics as in [35], which are based on geodesic distance ($\Delta(R_1, R_2) = \|log(R_1^T R_2)\|_F/\sqrt{2}$) over the manifold of rotation matrices between ground truth and predicted 3D viewpoints (azimuth, elevation and in-plane rotation). For more details of the metric definition, please refer to [35]. From the results, it is clear that our methods significantly outperforms the baseline CNN.

To evaluate the effect of synthetic data versus real data, we compare model trained with real images (Ours-Real) and model trained with rendered images (Ours-Render). For Ours-Real, we flip all VOC12 train set images to augment real training data. For Ours-Render, we only use synthetic images for training. In Figure 6, we see an 32% median azimuth error decrement from Ours-Real (23.5°) to Ours-Render (16°). By combining two data sources (we feed the network with both real and rendered images), we

	aero	bike	boat	bottle	bus	car	chair	table	mbike	sofa	train	tv	mean
$Acc_{\frac{\pi}{6}}$ (Tulsiani, Malik)	0.78	0.74	0.49	0.93	0.94	0.90	0.65	0.67	0.83	0.67	0.79	0.76	0.76
$Acc_{\frac{\pi}{6}}$ (Ours-Render)	0.74	0.83	0.52	0.91	0.91	0.88	0.86	0.73	0.78	0.90	0.86	0.92	0.82
$MedErr$ (Tulsiani, Malik)	14.7	18.6	31.2	13.5	6.3	8.8	17.7	17.4	17.6	15.1	8.9	17.8	15.6
$MedErr$ (Ours-Render)	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

Table 2. **Viewpoint estimation with ground truth bounding box.** Evaluation metrics are defined in [35], where $Acc_{\frac{\pi}{6}}$ measures accuracy (the higher the better) and $MedErr$ measures error (the lower the better) based on geodesic distance over the manifold of rotation matrices. Model from Tulsiani, Malik [35] is based on TNet, a similar network architecture as ours except the loss layer. While they use real images from both VOC 12 train and ImageNet for training, Ours-Render only uses rendered images for training.

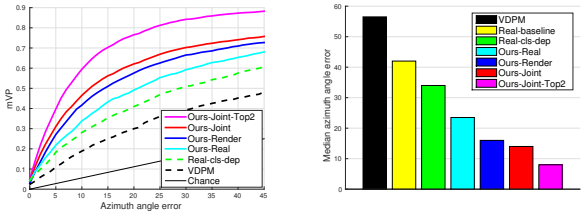

Figure 6. **Viewpoint estimation performance on detected object windows on VOC 2012 val.** *Left:* mean viewpoint estimation accuracy as a function of azimuth angle error δ_θ. Viewpoint is correct if distance in degree between prediction and groundtruth is less than δ_θ. *Right:* medians of azimuth estimation errors (in degree), lower the better.

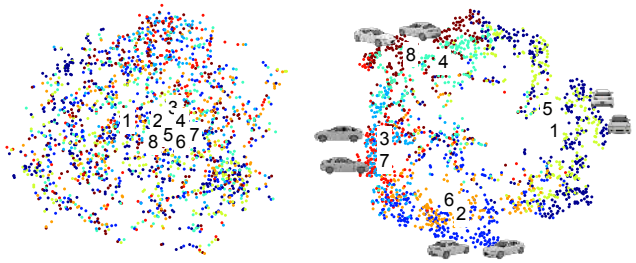

Figure 7. **Feature space visualization for R-CNN and our view classification CNN**. We visualize features of car images extracted by original R-CNN for detection (left) and our CNN for viewpoint estimation (right) by t-SNE. Each feature point of an image is marked in color corresponding to its quantized azimuth angle (8 bins for $[0, 2\pi)$). For points of same angle bin, their center is labeled on the plot by angle bin id.

get another 2° less error.

In Figure 6 right, we also compare Real-baseline (softmax loss and class-independent architecture) with Real-cls-dep (softmax loss and our class-dependent architecture) and Ours-Real (geometric aware loss and class-dependent architecture). We see both of our architecture choice and smooth loss function lead to better performance.

To show the benefits of fine-grained viewpoint estimation formulation, we take top-2 viewpoint proposals with the highest confidences in local area. Figure 6 left shows that having top-2 proposals significantly improve mVP when azimuth angle error is large. The improvement can be understood by observing ambiguous cases in Figure 10, where CNN gives two or multiple high probability proposals and many times one of them is correct.

5.3. Learned Feature Space Visualization

The viewpoint estimation problem has its intrinsic difficulty, due to factors such as object symmetry and similarity of object appearance at nearby views. Since our CNN can predict viewpoints well, we expect the structure of our CNN feature space to reflect this nature. In Figure 7, we visualize the feature space of our CNN (output of the last shared fully connected layer) in 2D by t-SNE [36], using "car" as an example. As a comparison, we also show the feature space from original R-CNN over the same set of images. We observe strong viewpoint-related patterns in our feature space: 1) images from similar views are clustered together; 2) images of symmetric views (such as 0° and 180°) tend to be closer; 3) the features form a double loop. In addition, as the feature point moves in the clock-wise direction, the viewpoint also moves clock-wisely around the car. Such observations exactly reflect the nature we discussed at the beginning of this paragraph. As a comparison, there is no obvious viewpoint pattern for R-CNN feature space.

5.4. Synthesis Parameter Analysis

In this section we show results of our control experiments, which analyze the importance of different factors in our "Render for CNN" image synthesis pipeline. The control experiments focus on the chair category, since it is challenging by its diversity of structures. We first introduce the five testbed data sets and the evaluation metrics.

Experimental Setup We refer to the test datasets using the following short names: 1) **clean**: 1026 images from the web, with relatively clean backgrounds but no occlusion, e.g., product photos in outdoor scenes. 2) **cluttered**: 1000 images from the web, with heavy clutter in the background but no occlusion. 3) **ikea**: 200 images of chairs photoed from an IKEA department store, with strong background clutter but no occlusion. 4) **VOC-easy**: 247 chair images

images per model	clean	clutter	ikea	VOC-easy	VOC-all	avg.
16	89.1	92.2	92.9	77.7	46.9	79.8
32	93.4	93.5	95.9	81.8	48.8	82.7
64	94.2	94.1	95.9	84.6	48.7	83.5
128	**94.2**	**95.0**	**96.9**	**85.0**	**50.0**	**84.2**

Table 3. **Effect of synthetic image quantity.** Numbers are $16V_{tol}$ (16 view accuracy with tolerance).

	clean	clutter	ikea	VOC-easy	VOC-all	avg.
nobkg	**95.4**	93.1	86.2	78.1	48.5	80.3
bkg (indoor)	94.2	91.6	92.8	80.6	48.9	81.6
bkg (all)	94.2	**95.0**	**96.9**	**85.0**	**50.0**	**84.2**

Table 4. **Effect of background synthesis.** Numbers are $16V_{tol}$ (16 view accuracy with tolerance).

from PASCAL VOC 12 val, no occlusion, no truncation, non difficult chair images. 5) **VOC-all**: all 1449 chair images from PASCAL VOC 12 val. While the clean and cluttered sets exhibit a strong non-uniform viewpoint distribution bias, the VOC-easy and VOC-all set have a similar tendency with weaker strength. The ikea dataset has close-to-uniform viewpoint distribution. All images are cropped by ground truth bounding boxes. The groundtruth for the clean, cluttered and ikea dataset are provided by the authors, those for VOC-easy and VOC-all are by PASCAL 3D+. Usage of these five datasets instead of just one or two of them is to make sure that our conclusion is not affected by dataset bias.

Unless otherwise noted, our evaluation metric is a discrete viewpoint accuracy with "tolerance", denoted as $16V_{tol}$. Specifically, we collect 16-classes azimuth viewpoint annotations (each of which corresponds to a 22.5° slot) for all the data sets we described above. As for testing, if the prediction angle is within the label slot or off by one slot (tolerance), we count it as correct. The tolerance is necessary since labels may not be accurate in our small scale human experiments for 16-classes viewpoint annotation[3].

Effect of Synthetic Image Quantity We separately fine tune multiple CNNs with different volumes of rendered training images. We observe that the accuracy keeps increasing as the training set volume grows (Table 3) from $6928 * 16 = 111K$ images to $6928 * 128 = 887K$ images. The observation confirms that quantity matters and more training data from synthesis does help the training of the CNN.

[3]Accurate continuous viewpoint labels on PASCAL 3D+ are obtained by a highly expensive approach of matching key points between images and 3D models. We do not adopt that approach due to its complexity. Instead, we simply ask the annotator to compare with reference images.

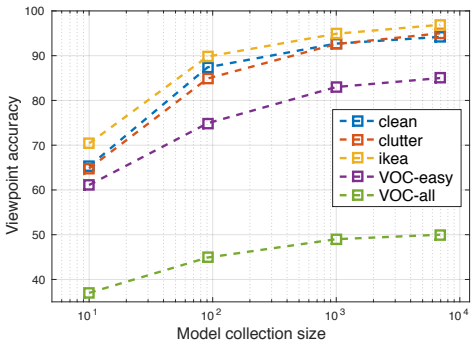

Figure 8. **Effect of 3D model collection size.** Vertical axis numbers are $16V_{tol}$ (16 view accuracy with tolerance).

Effect of Model Collection Size We keep the total number (887k) of rendered training image fixed and change the number of 3D models used for training data synthesis. In Figure 8 we see that as the model collection size increases, system performance continually increases and tends to plateau as number of models approaches 10^4.

Effect of Background Clutter As objects in the real world are often observed in cluttered scenes, we expect the network to perform better when trained on images with synthetic backgrounds. In Table 4, we see that *nobkg* (trained on rendered images with no background, i.e., black background) performs worse than *bkg* (trained on rendered images with synthetic backgrounds - cropped images from SUN397 scene database) in the ikea, VOC-easy and VOC-all data sets, which are with cluttered backgrounds. We also notice that *nobkg* performs better in the clean data set. This is reasonable since *nobkg* network has been working hard on clean background cases. Another discovery is that *bkg-indoor* (background restricted to indoor scenes of 177 categories), while surpassing *nobkg* , does not match *bkg-all* (no restriction on background). Possible explanation is that higher diversity of background helps minimize overfitting.

5.5. Qualitative Results

Figure 9. **3D model insertion.** Recovered 3D viewpoint reduces search space of model retrieval.

Besides azimuth estimation, our system also estimates elevation and in-plane rotation of the camera. To visualize

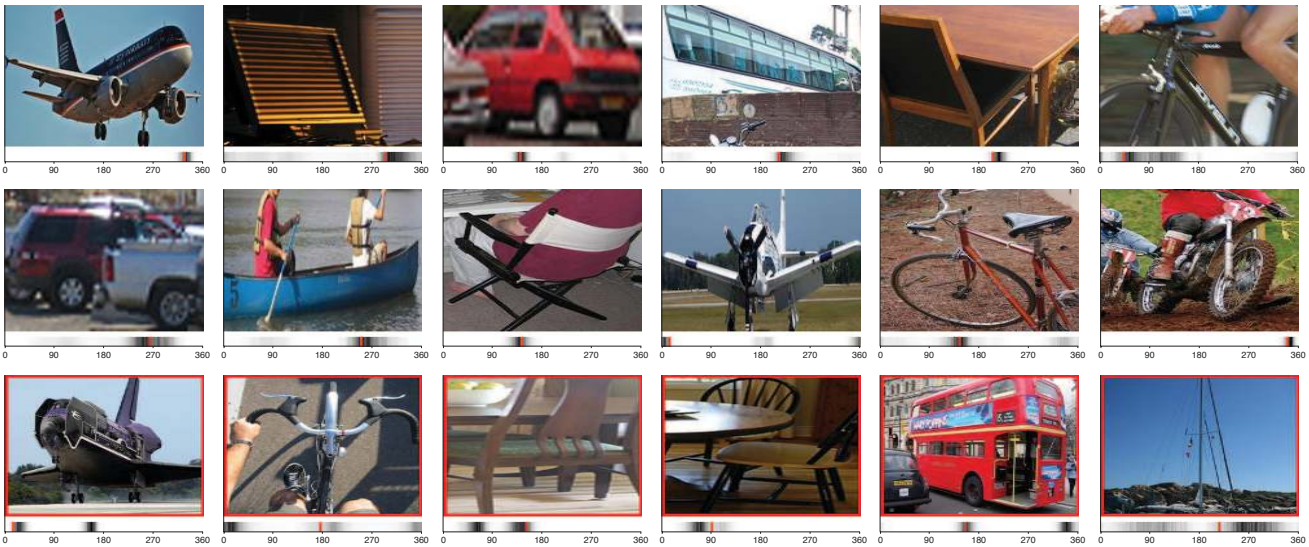

Figure 10. **Viewpoint estimation example results.** The bar under each image indicates the 360-class confidences (black means high confidence) corresponding to $0^\circ \sim 360^\circ$ (with object facing towards us as 0° and rotating clockwise). The red vertical bar indicates the ground truth. The first two rows are positive cases, the third row is negative case (with red box surrounding the image).

this ability, Figure 9 shows examples by model insertion for objects detected by R-CNN. The inserted 3D models are searched from our library by similarity.

Figure 10 shows more representative examples of our system. For each example, we show the cropped image by a bounding box and the confidence of all 360 views. Since viewpoint classifiers are regularized by our geometry-aware loss and sharing lower layers, the network learns about *correlations among viewpoints*. We observe interesting patterns. First of all, for simple cases, our system correctly outputs a clear single peak. Second, for those challenging cases, even though our system may fail, there is still a lower peak around the ground truth angle, validated both by the examples and our experiment results presented in Figure 6. Higher level systems (e.g. 3D model alignment, keypoint detector) can use those viewpoint proposals to reduce search space and increase accuracy. This proposing ability is not available for a regression system.

We observe several typical error patterns in our results: occlusion, multiple objects, truncation, and ambiguous viewpoint. Figure 10 illustrates those patterns by examples. For cases of occlusion the system sometimes gets confused, where the 360 classes probabilities figure looks messy (no clear peaks). For cases of ambiguous viewpoints, there are usually two peaks of high confidences, indicating the two ambiguous viewpoints (e.g. a car facing towards you or opposite to you). For cases of multiple objects, the system often shows peaks corresponding to viewpoints of those objects, which is very reasonable results after all. See supplementary material for more results.

6. Conclusion and Future Work

We demonstrated that images rendered from 3D models can be used to train CNN for viewpoint estimation on real images. Our synthesis approach can leverage large 3D model collections to generate large-scale training data with fully annotated viewpoint information. Critically, we can achieve this with negligible human effort, in stark contrast to previous efforts where training datasets have to be manually annotated.

We showed that by carefully designing the data synthesis process our method can significantly outperform existing methods on the task of viewpoint estimation on 12 object classes from PASCAL 3D+. We conducted extensive experiments to analyze the effect of synthesis parameters and input dataset scale on the performance of our system.

In general, we envision our Render for CNN pipeline can be extended to many tasks beyond viewpoint estimation, especially to those that annotations are hard to acquire such as segmentation, dense correspondence and depth estimation. Furthermore, 2D images and 3D shapes/scans can be connected through our pipeline [18], thus information can be transported between the two domains bidirectionally.

Acknowledgement. This work was supported by NSF grant IIS 1016324 and 1546206, ONR grant N00014-13-1-0341, a Google Focused Research Award, a gift from Apple Corporation, the Mac Planck Center for Visual Computing and Communication and GPUs donated by NVIDIA Corporation. We are also thankful to the anonymous reviewers for their helpful comments and Tao Du, Manolis Savva, Yuke Zhu, Alexandre Alahi for valuable discussion.

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 *CVPR*, 2014.

[2] K. W. Bowyer and C. R. Dyer. Aspect graphs: An introduction and survey of recent results. *International Journal of Imaging Systems and Technology*, 2(4):315–328, 1990.

[3] C. M. Cyr and B. B. Kimia. A similarity-based aspect-graph approach to 3d object recognition. *International Journal of Computer Vision*, 57(1):5–22, 2004.

[4] P. David, D. Dementhon, R. Duraiswami, and H. Samet. Softposit: Simultaneous pose and correspondence determination. *International Journal of Computer Vision*, 59(3):259–284, 2004.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[6] Y. Deng, Q. Yang, X. Lin, and X. Tang. A symmetric patch-based correspondence model for occlusion handling. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1316–1322. IEEE, 2005.

[7] B. Falcidieno. Aim@shape. http://www.aimatshape.net/ontologies/shapes/, 2005.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[10] P.-L. George. Gamma. http://www.rocq.inria.fr/gamma/download/download.php, 2007.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR 2014*, 2014.

[13] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Inferring 3d object pose in rgb-d images. *arXiv preprint arXiv:1502.04652*, 2015.

[14] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing image style. *arXiv preprint arXiv:1311.3715*, 2013.

[15] J. J. Koenderink and A. J. Van Doorn. The singularities of the visual mapping. *Biological cybernetics*, 24(1):51–59, 1976.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[18] Y. Li, H. Su, C. R. Qi, N. Fish, D. Cohen-Or, and L. J. Guibas. Joint embeddings of shapes and images via cnn image purification. *ACM Trans. Graph.*, 2015.

[19] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1688–1695. IEEE, 2010.

[20] J. J. Lim, A. Khosla, and A. Torralba. Fpm: Fine pose parts-based model with 3d cad models. In *Computer Vision–ECCV 2014*, pages 478–493. Springer, 2014.

[21] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing ikea objects: Fine pose estimation. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2992–2999. IEEE, 2013.

[22] F. Massa, M. Aubry, and R. Marlet. Convolutional neural networks for joint object detection and pose estimation: A comparative study. *arXiv preprint arXiv:1412.7190*, 2014.

[23] J. Michels, A. Saxena, and A. Y. Ng. High speed obstacle avoidance using monocular vision and reinforcement learning. In *Proceedings of the 22nd international conference on Machine learning*, pages 593–600. ACM, 2005.

[24] N. Mitra, M. Wand, H. R. Zhang, D. Cohen-Or, V. Kim, and Q.-X. Huang. Structure-aware shape processing. In *SIGGRAPH Asia 2013 Courses*, SA '13, pages 1:1–1:20, New York, NY, USA, 2013. ACM.

[25] R. Nevatia and T. O. Binford. Description and recognition of curved objects. *Artificial Intelligence*, 8(1):77–98, 1977.

[26] N. M. Oliver, B. Rosario, and A. P. Pentland. A bayesian computer vision system for modeling human interactions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):831–843, 2000.

[27] X. Peng, B. Sun, K. Ali, and K. Saenko. Exploring invariances in deep convolutional neural networks using synthetic images. *arXiv preprint arXiv:1412.7122*, 2014.

[28] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3d2pm–3d deformable part models. In *Computer Vision–ECCV 2012*, pages 356–370. Springer, 2012.

[29] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007.

[30] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *ACM SIGGRAPH computer graphics*, volume 20, pages 151–160. ACM, 1986.

[31] O. Sorkine and M. Alexa. As-rigid-as-possible surface modeling. In *Proceedings of the Fifth Eurographics Symposium on Geometry Processing*, SGP '07, pages 109–116, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[32] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, volume 2, page 5, 2010.

[33] H. Su, M. Savva, L. Yi, A. X. Chang, S. Song, F. Yu, Z. Li, J. Xiao, Q. Huang, S. Savarese, T. Funkhouser, P. Hanrahan, and L. J. Guibas. ShapeNet: An information-rich 3d model repository. http://www.shapenet.org/. 2015.

[34] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 213–220. IEEE, 2009.

[35] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Computer Vision and Pattern Regognition (CVPR)*, 2015.

[36] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[37] M. A. O. Vasilescu and D. Terzopoulos. Multilinear independent components analysis. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 547–553. IEEE, 2005.

[38] Z. Wu, S. Song, A. Khosla, X. Tang, and J. Xiao. 3d shapenets for 2.5 d object recognition and next-best-view prediction. *arXiv preprint arXiv:1406.5670*, 2014.

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

[40] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010.