Learning 3D Object Orientations From Synthetic Images

Ruizhongtai (Charles) Qi
Stanford University
rqi@stanford.edu

Abstract

While information of 3D object orientation is very useful in image retrieval and scene understanding, large collection (at the scale of millions) of labeled training data is not available and expensive to obtain. This project makes use of annotated 3D shape models to synthesize millions of training images of various orientations and trains deep convolutional neural networks to robustly predict object orientations in real images under model variations, background clutters and complex lightning conditions. We show that it's feasible to transfer knowledge from synthetic images to real images and the model can achieve higher than 90% 16-view classification accuracy in several real image test sets.

1. Introduction

Motivation

Object orientation or viewpoint estimation is an important step for image retrieval and model matching. When searching an object model to match a 2D image, a good estimation of the object's pose, i.e. the viewpoint towards the object, can greatly reduce the search space. In scene understanding and reconstruction problems, it is also critical to accurately estimate the viewpoint to discover the 3D structure of the scene.

While it is not difficult to predict 3D object orientation with clean background and of the same objects in different poses, the problem quickly becomes much harder when we consider variations in object models, object size, lighting conditions and background clutters. The real-world scene complexity makes the problem nearly impossible to be solved by explicitly programming or building a model trained on only a few hundred images. However, large number (say hundreds of thousands of) of real images with accurate orientation annotations are not available and it would be very expensive to collect them.

Basic Approach

In this project, we will use an annotated large 3D shape data set (ShapeNet [1]) to generate rendered 2D images with accurate orientation labels. By using the annotated 3D shape data set, we can generate as many training images as we wish, thus solving the problem of lack of training data. We will use these rendered training images to fine tune a deep CNN pre-trained with ImageNet data [2]. In that way, we have low-level representations from ImageNet models and will train higher level features from rendered images.

There are two fundamental research questions for our approach. The first one is whether our model is able to transfer knowledge learnt from rendered images to natural images. The second question is whether CNN model is able to learn representations for geometry inference, e.g. 3D object orientation. Our results show that both are possible, the CNN is able to learn geometric features and the transfer learning from rendered images to natural images is very successful.

Figure 1. Challenge of knowledge transfer from 3D models to rendered images and then to real images.

Task Definition

We will make two simplifications of the view estimation problem. Firstly, while view estimation of rigid objects include both azimuth angle and altitude angle estimation, we focus on the azimuth angle and we will show that on altitude angle our approach can achieve similarly good results. Secondly, we formulate view estimation problem as a classification problem. Strictly speaking, viewpoint is a continuous attribute and it's more accurate to model our problem as a regression problem. However, since the focus of this work is on transferring knowledge from rendered images (from 3D shape models) to natural images and on using deep learning (CNN architecture) to extract geometry

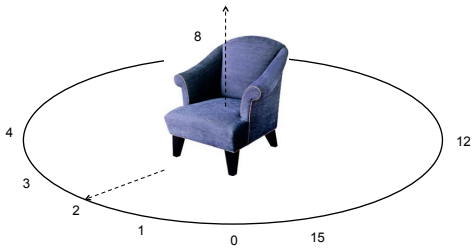

Figure 2. 3D object orientation estimation as a classification problem - azimuth angle is quantized to 16 classes from 0 to 15.

information, whether to formulate the problem as classification or regression does not affect us from studying these two aspects.

More specifically, we will focus on estimating azimuth viewpoint of chair objects as a 16-class classification problem. Azimuth angle is in the range of 0 to 360 degree and label 0 means the angle is between -11.25 to 11.25 degree, label 1 means the angle is between 11.25 to 33.75 degree etc. The reason that we look at chair objects is two fold. Firstly, we have more than 5000 thousand annotated 3D models for chair in our shape data set, which covers most of the common models seen in real world. Secondly, chairs have large variations in shapes and if we can do well on chairs we are probably able to do well on other types of objects as well.

Evaluation method

Notice that we will train our model on rendered images (from 3D shape models) and test it on natural images. We will evaluate our system by accuracy of viewpoint estimation (as a classification problem) on the test set. Since it is extremely hard for a human to accurately label the viewpoint of an image, we allow for a small error in labels when calculating the accuracy. We will consider the prediction correct if its distance to the label is zero or one.

We will also compare our results with traditional models and CNN models trained on natural images (much less then rendered images in number).

Contributions

Firstly, we verified the feasibility of using synthesized images (from large collection of 3D models) to train models for testing on real images. It opens a new door to a under-explored research field. Secondly, we showed that CNN is capable of geometric inference (predict object orientation). Based on detected object type and bounding box, we can achieve very accurate orientation estimation (more than 90% classification accuracy) of chairs in real scene images with all kinds of model variations, clutters, untight bounding boxes and complicated lighting conditions. Lastly, error analysis and feature visualization provide us with insight into valuable future research areas such as CNN regressor and 3D scene synthesis.

2. Related Work

Recently there are two hot topics in computer vision community: one is deep convolutional network as a powerful model and the other is 3D vision that involves problems related to geometric information of objects in images. This project tends to connect the two topics by trying to solve a 3D vision task, estimation 3D object orientations or viewpoint, by deep convolutional neural networks. More importantly we study how synthetic images can be used for training and what their limits are. The related works are organized by sub-topics in the following.

Synthetic Images for Training: While most data-driven vision projects use real images to train their model, researchers have also tried to use synthetic images to make up for lack of training images in specific viewpoints [8]. However, current work requires human to pick similar 3D models of the objects in images and manually alignment the model to the image, which greatly limits the amount of training images they can synthesize. While other group [9] tried to automatically synthesize more images, the experiments are restricted to a small number of 3D models thus have low level of generalization capability. Our project will have both automatic synthesis procedure and use a large collection of 3D models that empower us to scale up the system with high generalization ability.

Object Orientation Estimation: Object orientation is an important geometric feature of the objects in images and can be important for image retrieval and 3D reconstruction. While previously works like [10] [4] approaches the problem as a regression problem this project will simplify orientation as a quantized value thus model it as a classification problem. Also, some work like [7] has combined orientation estimation with object detection or image retrieval, due to the time limit, this project will focus on the orientation problem alone (higher level application can use orientation prediction result as a known fact then).

Using 3D Shape Models: Since more 3D shape models are available now and they can act as a good source of prior knowledge for computer vision tasks, projects such as [8] [12] [3] have used 3D models. However, none of them directly uses 3D models as a source of training data. This project will take the more bold and innovative path to render images from 3D models for model training.

Convolutional Neural Networks: Ever since the landmark paper introducing AlexNet for object classification [6], deep convolutional neural network has played a major role in pushing the state-of-the-art for vision tasks. Recently there have also been many papers using ConvNet to infer geometric knowledge such as pose, depth and normal

surfaces of the objects [3] [2] [11], this project follows this line to explore the ability of ConvNet in estimating object orientations.

3. Approach

We have two powerful tools. First, we will use deep learning (deep convolutional nets specially) as a powerful representation learning tool. Second, we will make use of a large 3D shape data set (ShapeNet) to generate large volume (say 1M) of rendered images for training. Since we have thousands of 3D models for certain types of objects (e.g. chairs), we have full control of how the training data look like in terms of depth, lighting, viewpoint, clutters and occlusion, and accurate viewpoint labels are just available for free. Another benefit of rendered images is that we can have even number of images of different views while natural images are severely biased in viewpoint distribution (thus naively training on real images tends to fail if test set is not of the same distribution).

We train our model for chair viewpoint estimation using rendered images and test our model on real world chair images. This is transfer learning. A lot of effort is on studying how to maximally transfer knowledge from training set to test set is expected.

3.1. Network Fine-tune

We use a fine-tune approach, i.e. fine tuning fully connected layers (and also the last few conv layers if we render enough training images) of a pre-trained ConvNet model on ImageNet data. We will train our model with Caffe [5] and GPUs.

As seen in Figure 3, we start from a classical ConvNet architecture (RCNN model from Caffe Model Zoo which has 5 conv layers and 3 FC layers plus one Softmax loss layer) with weights trained from ImageNet data set. Then we fix lower layers of the network and fine tune higher layers with our rendered images and viewpoint labels. Finally, we test our system on real world images.

Several comments should be made here. First, we choose RCNN (similar as AlexNet) architecture since it's well known and is the most commonly used one, which not necessarily means it's the best choice. We would rather pay more attention to the image synthesis and error analysis than to network architecture tuning for this project. Second, we have also experimented with regression model, which is more tricky since orientation is periodic. We used periodic L1 and L2 norms and the regression models achieves similar performance as the classification model, so we will focus on the classification model here.

3.2. Image Synthesis

For image synthesis we start from 3D object models. The first step is to render object images of different orientations.

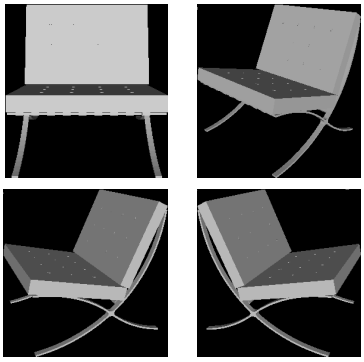

Figure 4. A sample of rendered images from baseline setting.

During the rendering process we may need to control various rendering conditions such as texture, lighting etc. The second step is to perturb the pure rendered image with background clutter, cropping etc. In following paragraphs, we will show step by step how we discover the strategies of image synthesis that works best for object orientation estimation task.

During the discussion, we always use the same 5057 annotated 3D chair models for image rendering. We use blender for the rendering and synthesis steps.

First try

At the first time we approach the problem, for each model, we generate 16 gray-scale images of the model from a fixed set of views. Altitude angle is fixed at 25 degree, which results in around 80K rendered images. In terms to lighting condition, we have 4 fixed point light sources on a sphere. The background for rendered images is clean (uniform intensity).

This first attempt gives us a lower bound of our system performance and we will point later that there are large room of improvement in terms of training data generation. By careful data perturbation and various data augmentation tricks we can push the system performance to a much higher level.

Random light

We notice that when we fix positions of lighting sources, there is a strong pattern of brightness (e.g. bright chair backs and dark chair seats), as can be seen in Figure 4. When a pattern recognition model (here, the ConvNet) sees these training data, it tends to quickly pick up the patter of lighting. While this pattern consistently works on all training images that is rendered under the same lighting set up, it will fail in real world images where lighting conditions vary a lot. Thus as a step to increase our system's robustness, we use random positions for four lighting sources.

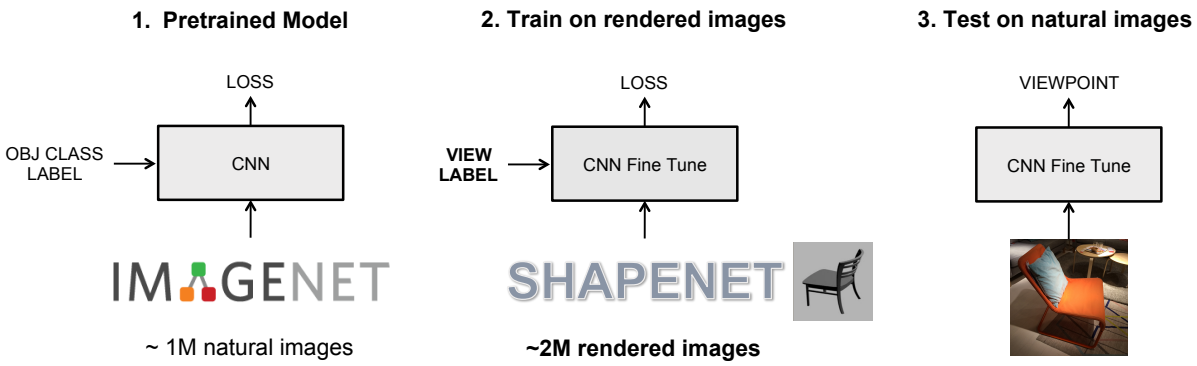

Figure 3. Fine tune CovNet trained from ImageNet data. Transfer knowledge learnt from rendered images to natural images.

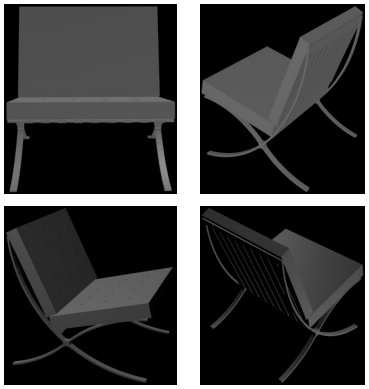

Figure 5. A sample of rendered images using random light sources.

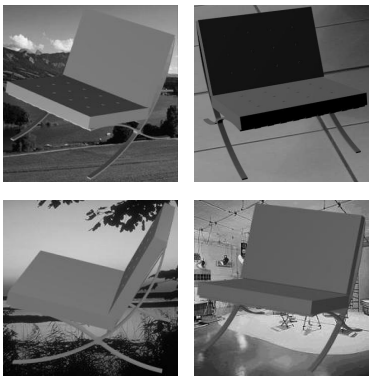

Figure 6. A sample of rendered images using random light sources and with synthetic clutter background.

Now the lighting sources will be uniformly distributed on a sphere. In results part, we will see that by using random light sources we can have a large accuracy gain compared with using images using fixed lighting sources.

Background clutter

We also notice that since the rendered images in 5 has clean background there are always sharp contours of objects. Also, for real world images there are often background clutter that will misguide our classifier. Thus we decide to synthesis background to the rendered images. In Figure 6, we can see that the training images now look much more like real world ones. We will also see in the results part that by adding clutter background, we again achieve a big accuracy gain. We used scene images fron SUN database for backgrounds.

Mixed background

Since clean background is just a special type of background, we can also mix clean and cluttered background to achieve a richer background distribution. Furthermore, though not done here, we can include multiple background image sources covering indoor, outdoor and other many cases.

Data augmentation

For the previous steps we keep the number of entire training images fixed at around 80,000. Using these images we are only able to open up the top FC7 and FC6 layers of the ConvNet, otherwise the model will quickly overfit too much on the training data. At this step we decide to generate much more synthetic training images by perturbing the images from multiple perspectives. We vary altitude angles in a wider range, sample more azimuth angles for each model, crop the images so that the bounding box is untight, synthesize more backgrounds composing of different colors and images. At last we get more than 2 million training images, a sample of them are shown in Figure 7. We can then open up more layers for fine-tuning. We tried different fine-tuning scale from opening all layers to open only the

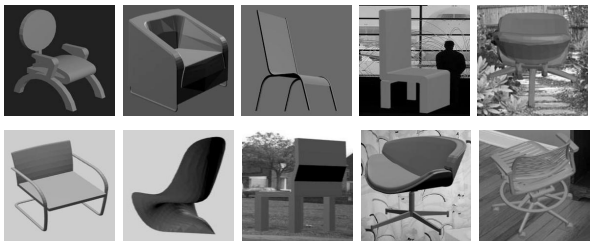

Figure 7. A sample of final collection of synthetic images for training (only 10 model images out of 5057 models and 2 millions images are shown here).

top FC7 layer and will report the result of the best of them.

4. Experiment Results

4.1. Data set

We use 5057 3D chair models with annotations on viewpoint/orientation (a sample of them can be seen in left most picture in Figure 1), thus we can render as many as 2D images as we like. For example, if for the 5000 models, we render 16 images of 16 different views for each model, we will have 80,000 rendered images already beyond the size of any viewpoint estimation data set stated in literature. Moreover, since we have full control of the rendering step and we have viewpoint annotations for our 3D models, the labels for rendered images will come for free. Those rendered images will be used as training and validation data set.

We use blender for rendering and it takes around one day to render and synthesize backgrounds for 2 million images. Specifically rendering parameters: we use altitude angles uniform distributed between -10 to 60 degrees, we sample azimuth angles around 16 points between 0 to 360 degrees, objects depth are varied uniformly between 5 and 7, four point lighting sources are used. All backgrounds are either from SUN database or are uniform gray scale from 0 to 255. Since the object orientation has weak relation with colors all images we render are in gray scale (for cases color matter, for example cars, we can also render colorful images).

For test data set, we have collected more than three thousands of natural scene images of chairs in different viewpoints. These real world images have also been categorized to various difficulty levels from the easiest ones with clean background to difficult ones with occlusions and complex backgrounds (see Figure 8). We have manually labeled them into 16 classes of quantized viewpoints.

4.2. Training

Each time we train a model, we firstly search the hyperparameter space to find a set of reasonable learning rate and regularization weights. We use a single Tesla 42C GPU to train the network and it takes around 6 to 12 hours to achieve convergence. Learning curves are plotted to ensure

Figure 8. Test data sets. The first 2 data sets are of very biased distributions - more chairs are facing towards you then facing opposite to you while the IKEA data set is of nearly uniform orientation distribution.

that the network does not overfit too much to the training data.

4.3. Results

In this section, we show the results of our 3D object (on chairs specifically) orientation estimation system. One important note is that instead of heavily tuning the network architecture and hyperparameters, our project focuses more on designing training data to teach the network to learn essential and robust features for object viewpoint. The results we present are from tuned models for the corresponding data set.

For system evaluation, we have 4 test data sets which are images with simple/clean background, images with clutter background, images from IKEA shops with more uniform distributions in terms of viewpoint and images with untight bounding boxes. Thus there are 3 testing set (clean, clutter and IKEA) with tight bounding boxes and one data set with untight bounding box (IKEA-U). There are 2 testing sets (clean, clutter) of biased viewpoint distributions (much more images are chairs facing towards camera) and 2 testing sets of unbiased viewpoint distribution (IKEA, IKEA-U).

To compare with model trained on real images, we set up a baseline by fine tuning the model using five hundred real images. While more real training data can be collected the

	clean	clutter	IKEA	IKEA-U	average
Baseline	80.8	84.8	50.5	44.4	**65.1**
FirstTry	54.4	67.5	35.7	28.6	**46.6**
RandLight	90.4	85.4	70.4	50.0	**74.0**
ClutterBg	84.5	86.1	89.8	72.4	**83.0**
MixedBg	87.6	87.8	92.9	74.0	**85.6**
DataAug	**90.5**	**91.8**	**97.5**	**92.9**	**93.2**

Table 1. Orientation estimation accuracy under different training data settings. Baseline is using 500 real images for training. FirstTry, RandLight, ClutterBg and MixedBg use 80,000 synthetic images and DataAug uses 2 millions images, each of which corresponds to the incremental image synthesis setting discussed in section 3.2.

effort of taking training data that has magnitude higher than five hundred is too large to be feasible in this short project period. Nevertheless, by comparing results of different rendering conditions and this baseline we can still gain much insights. Also the lack of real training images validate the advantage of using synthetic images for training.

In Figure 1, we see that by simply introducing random lighting we can gain a 27.4 points of accuracy improvement and by adding clutter background we gain another 9 points boost. This tells us an important lesson that it is critical to introduce variances i.e. data perturbations to aspects that irrelevant to the goal of viewpoint estimation. In that way, we can teach the ConvNet model to focus on more robust features and not pay too much attention to less relevant or unreliable patterns such as lighting or sharp contours.

Furthermore, by mixing the background we get 2.6 more points and by data augmentation we get the last 7.6 more points in average accuracy. In the data augmentation setting we opened up conv4, conv5 and fc6, fc7 layers of the model and observes little overfit thanks to the large number of 2 million training images. It's worth noticing that since data augmentation introduces training images with untight bounding boxes, we can see a nearly 20% accuracy improvement in IKEA-U (uncropped) data set from MixedBg to DataAug.

In Figure 9, we can see some positive examples from the cluttered background test set. We can see that the testing images are not that simple - strong cluttered backgrounds with strong edge patterns and complex lightning conditions. To verify that the network really learns to separate chair objects of different orientations, we visualize the top feature layer's outputs by t-SNE (Figure 10). We can see the features of images with 0 to 15 orientation class lie in a circle and can be separated.

4.4. Error analysis

To understand why the model fails on the rest images, we dive deep into mistakenly classified images in the clean

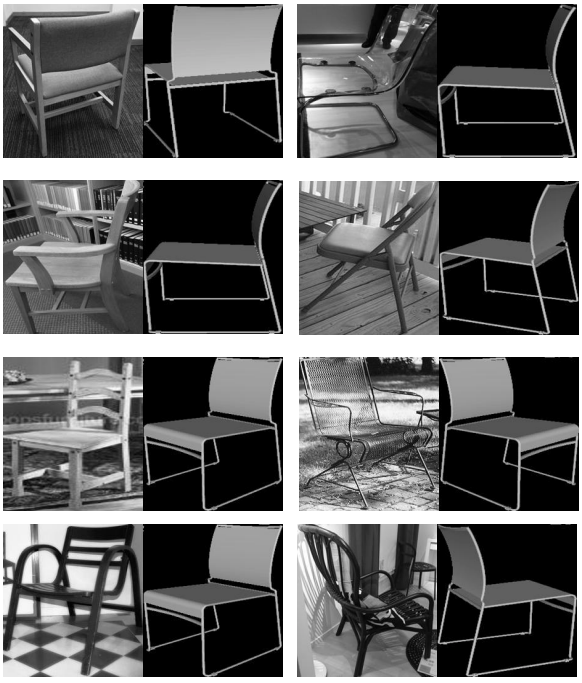

Figure 9. Positive examples (orientation classified correctly). For each small figure, the left side is the input image to the model, the right side is how our model thinks the chair is orientated - you can observe consistence of orientations in all 8 cases shown here.

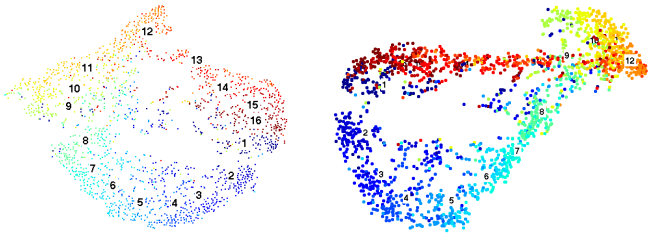

Figure 10. Visualization of CNN's last layer output - CNN features of a set of validation synthetic images. The left is 2D visualization and the right is 3D visualization.

background data set. We have observed three major error patterns (shown in Figure 4.4). The first pattern is **ambiguous views**, for example the front-back views and images with around 0 altitude angle can be hard to distinguish, even for humans. To make those images correct, we may need to push the network further to learn detailed parts of the object, e.g. handles or object-self occlusion patterns. The second pattern is that images of **unseen models** are harder to classify correctly. For examples barber and throne chairs are not present in the 3D model base, thus the network has less knowledge of these objects. As the 3D model market keeps growing we expect to see better coverage of 3D models thus less unseen model types. Another pattern is that images with **low resolution** (less than 100 by 100) have much

Figure 11. Ambiguous views account for around 50% of errors.

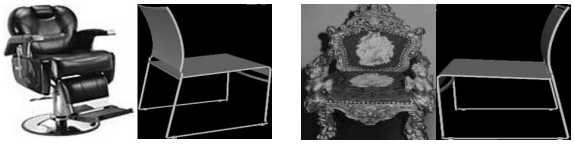

Figure 12. Accuracy much lower for unseen models.

Figure 13. Accuracy much lower for low resolution images.

lower accuracy since the details of the images are averaged out and the network will make decisions based more on the contours of objects. We expect that actively changing image resolution during training may help ease the problem.

5. Conclusion

In summary, in this project we have shown that it's feasible and actually very effective to use synthetic images to train a 3D object orientation estimator. Also we show that the ConvNet is capable of geometric inferences such as orientation prediction. This pipeline of synthesizing images from 3D model database, fine-tune with synthesized images and test on real images can be applicable to many other tasks such as depth estimation, scene layout prediction and object dimension estimation etc.

For future works, there are many directions to go. We can consider to synthesize object with occlusions. Although we have object-self occlusions, we do not have object-object occlusions in our training data, so the performance on occluded case is not very satisfactory. While simple ideas of adding random occlusion does not work well, we can try to synthesize scenes and generate occluded images from the larger scene. Another future direction is to apply our orientation learning pipeline to other object class e.g. cars and planes and maybe also predicting the altitude angles. Also, it's worthwhile to build a regression system on CNN features. One candidate loss that captures periodicity of orientation angles is a shifted cosine function.

Acknowledgement

The author sincerely acknowledge Hao Su in Stanford Geometric Computing Group for his supervision of the whole project and for his help in rendering the images. The author also wants to express great appreciation to the CS231N course staff for their great teaching and preparation of projects and notes. This is an exciting and rewarding class that introduces many cutting edge technologies and highly interesting applications.

References

[1] Shapenet: A large-scale 3d shape database, 2015.

[2] 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.

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

[4] C. Herdtweck and C. Curio. Monocular car viewpoint estimation with circular regression forests. In *Intelligent Vehicles Symposium (IV), 2013 IEEE*, pages 403–410. IEEE, 2013.

[5] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] 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.

[8] K. Rematas, T. Ritschel, M. Fritz, and T. Tuytelaars. Image-based synthesis and re-synthesis of viewpoints guided by 3d models. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3898–3905. IEEE, 2014.

[9] A. Rozantsev, V. Lepetit, and P. Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 2015.

[10] A. Saxena, J. Driemeyer, and A. Y. Ng. Learning 3-d object orientation from images. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 794–800. IEEE, 2009.

[11] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. Panda: Pose aligned networks for deep attribute modeling. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 1637–1644. IEEE, 2014.

[12] M. Z. Zia, M. Stark, and K. Schindler. Are cars just 3d boxes? jointly estimating the 3d shape of multiple objects. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3678–3685. IEEE, 2014.