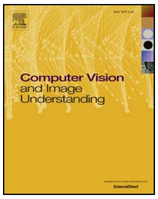
Computer Vision and Image Understanding 169 (2018) 75–89

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Computer Vision and Image Understanding

Viewpoint refinement and estimation with adapted synthetic data

Pau Panareda Busto*[,a,b], Juergen Gall[b]

[a] Airbus Group Innovations, TX4-ID, Munich, Germany
[b] University of Bonn, Computer Vision Group, Bonn, Germany

ARTICLE INFO

Keywords:
Domain adaptation
Pose estimation
Synthetic data

ABSTRACT

Estimating the viewpoint of objects in images is an important task for scene understanding. The viewpoint estimation accuracy, however, depends highly on the amount of training data and the quality of the annotation. While humans excel at labelling images with coarse viewpoint annotations like front, back, left or right, the process becomes tedious and the quality of the annotations decreases when finer viewpoint discretisations are required. To solve this problem, we propose a refinement of coarse viewpoint annotations, which are provided by humans, with synthetic data automatically generated from 3D models. To compensate between the difference between synthetic and real images, we introduce a domain adaptation approach that aligns the domain of the synthesized images with the domain of the real images. Experiments show that the proposed approach significantly improves viewpoint estimation on several state-of-the-art datasets.

1. Introduction

In order to estimate the viewpoint of objects in images precisely, an accurate annotation of the training data is required. Humans, however, perform poorly for estimating the viewpoint of an object accurately as illustrated in Fig. 1. Instead of annotating real images, synthetic data can be generated using 3D models (Marín et al., 2010; Mottaghi et al., 2015; Pishchulin et al., 2011; Sun and Saenko, 2014; Vázquez et al., 2014; 2011). While synthetic data provides accurate viewpoints, it either lacks the realism of real images or it is very expensive to generate. In particular, collecting a large variation of textured 3D shapes and combining them with coherent background scenes and illumination conditions is time-consuming.

We address this issue by leveraging human annotators and synthetic data, as depicted in Fig. 2, to avoid manual annotation by humans of fine viewpoints, which is time-consuming and erroneous, and to avoid the synthesis of a realistic dataset that captures the variations of real images, which is time and memory consuming. To this end, we ask humans to annotate only four coarse views, sketched in Fig. 3(a), and introduce an approach that refines the labels using synthetic data. Since synthetic data and real images belong to different domains as illustrated in Fig. 3(b), a domain adaptation approach is used for the refinement. General domain adaptation approaches like (Gong et al., 2012; Hoffman et al., 2013), however, are not sufficient for label refinement since they fail to distinguish viewpoint rotations by 180°. We therefore present a task-specific approach that takes advantage of the coarse labels of the real training samples.

A preliminary version of this work appeared in Busto et al. (2015). While the approach in Busto et al. (2015) was limited to cars, we extend the method to other categories and provide a thorough experimental evaluation. We also evaluate our approach with state-of-the-art features extracted from convolutional neural networks (CNN) (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014) and study the effect of truncated and occluded object instances. In addition, we also show how the refined datasets are able to obtain in some cases comparable or even better results than annotated training data with full human supervision. The evaluation, which is performed on six datasets for viewpoint estimation, reveals that our approach outperforms state-of-the-art domain adaptation methods.

2. Related work

2.1. Viewpoint estimation

Methods for viewpoint estimation are often based on popular object class detectors (Dalal and Triggs, 2005; Felzenszwalb et al., 2010; Girshick et al., 2014; Leibe et al., 2004) and learn a discrete set of pose classifiers. In Liebelt and Schmid (2010), Fidler et al. (2012), Pepik et al. (2012) and Hejrati and Ramanan (2014), annotations from 2D images are enhanced with 3D metadata to formulate 3D geometric models. On the contrary, Gu and Ren (2010) learns a mixture-of-templates that inherently captures the characteristics of projected views and Ozuysal et al. (2009) refines the hypothesis of 16 viewpoint detectors from 2D images with additional view specific Naïve Bayes classifiers.

* Corresponding author at: Airbus Group Innovations, Department of Data-Driven Technologies, Munich, Germany.
E-mail addresses: s6papana@uni-bonn.de, pau.panareda-busto@airbus.com (P. Panareda Busto), gall@iai.uni-bonn.de (J. Gall).

https://doi.org/10.1016/j.cviu.2018.01.005
Received 17 January 2017; Received in revised form 20 August 2017; Accepted 14 January 2018
Available online 31 January 2018
1077-3142/

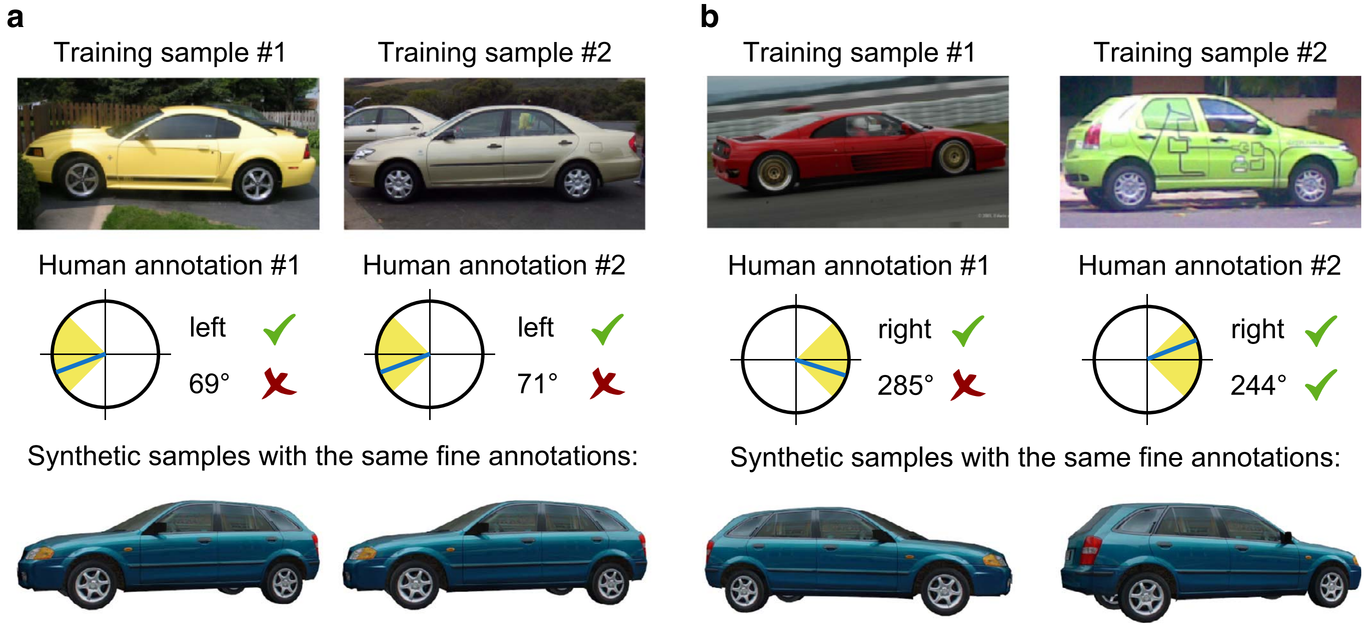


Fig. 1. Faulty annotations of fine viewpoints are introduced in human-annotated training datasets. While coarse labels like left or right are correct, the viewpoint annotations in degrees are not precise (a) and sometimes inconsistent (b) samples and fine annotations are taken from the Pascal3D+ dataset (Xiang et al., 2014).

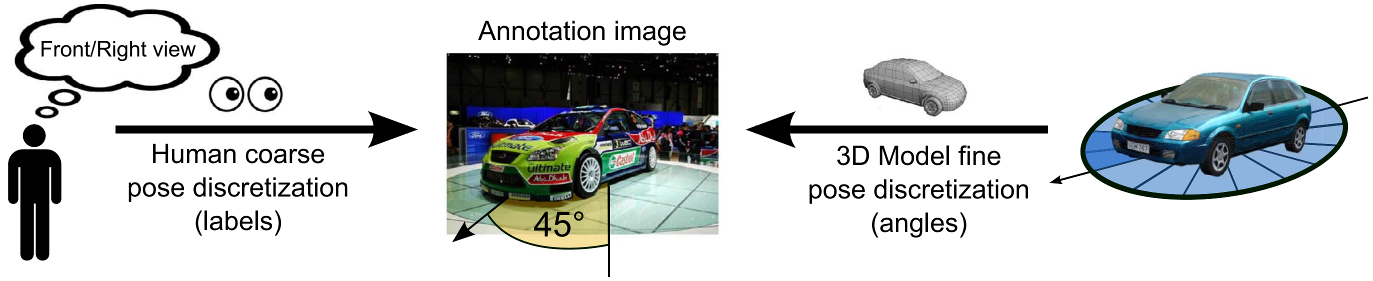
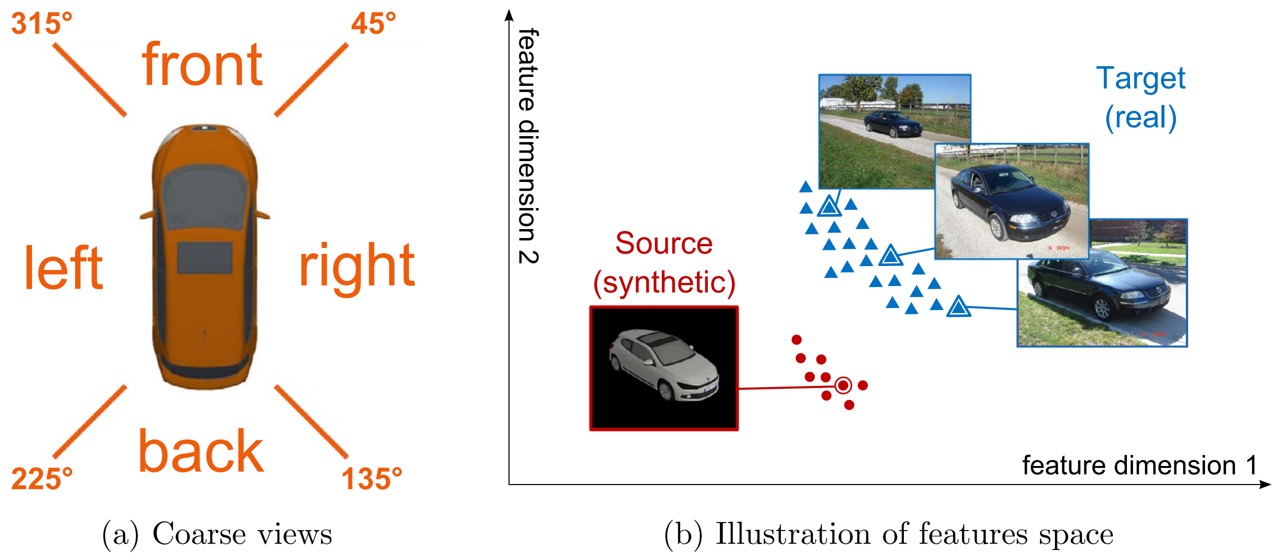


Fig. 2. Humans are perfect for annotating coarse viewpoints of objects in real images, but fail to estimate pose accurately at a fine level. 3D graphic models can be used to synthesize data at very accurate fine angles, but it is time-consuming to model all appearance variations present in real images. We therefore propose to leverage the abilities of humans of estimating coarse viewpoints and the pose accuracy of synthetic data.



(a) Coarse views

(b) Illustration of features space

Fig. 3. (a) The four views available for real images. (b) Synthetic and real images with the same annotated viewpoint lie in different domains within the feature space.

More recently, CNNs for object classification (Krizhevsky et al., 2012) have been retrained using 2D pose annotations in order to provide viewpoint probabilities as output channels coupled with the object class probability (Pepik et al., 2015; Tulsiani and Malik, 2015). In the study pursued in Ghodrati et al. (2014), simple frameworks that extract features from 2D bounding boxes with powerful encoders provided the same or even better viewpoint accuracies than state-of-the-art methods based on complex 3D models.

In contrast to classification approaches, regression approaches (Fenzi et al., 2013; Torki and Elgammal, 2011) do not require a discretisation of the viewpoints. In He et al. (2014), the viewpoint regression is integrated into a joint discriminative continuous parameterised model. The localisation and the continuous pose of objects are jointly estimated in Redondo-Cabrera et al. (2014) using a Hough forest regression voting scheme. Accumulated votes in the Hough space are later refined with a kernel density estimator to consolidate votes in a local region close to the current maxima. Similarly, Hough forests have been used for head pose estimation (Fanelli et al., 2013) where patches from depth images are used. In Glasner et al. (2012) a voting process is also used to refine the prediction of discretised pose classifiers. Recently, the study (Massa et al., 2016) concluded that CNNs for viewpoint classification outperform CNNs for viewpoint regression by a considerable margin. For further details on joint object detection and pose estimation, we refer to the studies Massa et al. (2014) and Elhoseiny et al. (2016).

2.2. Synthetic data

The use of synthetic images from rendered models and scenes as training data started to gain attention in the context of pedestrian detection. While Marín et al. (2010) only uses synthetic data generated from a popular game engine, Pishchulin et al. (2011) combines real with synthetic data from highly accurate 3D reconstructed humans. Both methods, however, do not consider the 3D information and collect only 2D images with automatically annotated bounding boxes.

Previously, the 3D spatial information of graphics models was already addressed in several works to estimate the viewpoint of object instances, as well as its localisation (Liebelt and Schmid, 2010; Mottaghi et al., 2015; Pepik et al., 2012; Schels et al., 2012; Stark et al., 2010; Zia et al., 2013). These algorithms are computationally expensive, since the object geometry is used to learn the spatial 3D relations of parts or features. In contrast to these works, we use the rendered models to synthesize training images with accurate viewpoint annotations. Instead of rendering 3D data, synthetic data can also be generated by defining a parametric model for synthesizing geometric shapes from a particular object class, used in both recognition and reconstruction, as proposed by Hejrati and Ramanan (2014).

Recently, Su et al. (2015) and Peng et al. (2015) tested the impact of synthetic data in CNNs by training millions of synthesized images from 3D models. Thus, the main challenge becomes the generation of extremely large amounts of data with as much intra-class variation as possible, e.g., viewpoint and shape, to avoid over-fitting. 3D models have also been used to annotate datasets (Matzen and Snavely, 2013; Xiang et al., 2014) by manually superposing them on top of 2D object instances. While the 3D models support humans and improve the accuracy of the annotation, the annotation process with 3D models is very slow and still prone to annotation errors. Instead of using synthetic data, Sedaghat and Brox (2015) proposed a supervised approach that automatically annotates cars in videos by bounding boxes and azimuth angles using structure from motion.

2.3. Domain adaptation

Domain adaptation addresses the problem when the training and test data are at least partially from different domains. To this end, either a transformation of the domains is estimated before the training of a classifier (Baktashmotlagh et al., 2013; Csurka et al., 2016; Gong et al., 2012; Gopalan et al., 2011) or the so-called source domain is used to regularize the learning of a classifier on the target domain (Jhuo et al., 2012; Pan et al., 2011). A popular choice in this context are support vector machines (Aytar and Zisserman, 2011; Duan et al., 2012; Hoffman et al., 2013; Xu et al., 2014; Yang et al., 2007). The approaches that estimate the transformations without a classifier like the geodesic flow kernel (Gong et al., 2012) learn mappings from the source and target domain into a joint, low-dimensional space. This can be done in an unsupervised manner where the target domain is unlabelled, or in a supervised or semi-supervised setting where the data from the target domain contains a few labelled samples. While these methods assume that the source and target domains are known, Gong et al. (2013) minimise the distance between latent domains, rearranging clusters of the annotated classes based on feature similarities. In contrast to these works, we use domain adaptation in a weakly supervised setting where only coarse labels are available for the training images of the target domain.

During the last years, domain adaptation methods focused on the optimization process for the domain alignment, where additional constraints for the optimization have been proposed (Aytar and Zisserman, 2011; Duan et al., 2012; Hoffman et al., 2013; Saenko et al., 2010; Xu et al., 2014). For instance, orthogonality constraints have been suggested for the transformation matrix (Baktashmotlagh et al., 2013; Jhuo et al., 2012), as well as relaxation techniques to make the optimization solvable (Gong et al., 2012; Xu et al., 2014). Other approaches, on the contrary, excel by its speed and simplicity. Fernando et al. (2013) computes a subspace alignment between domains in closed form and Sun et al. (2015) aligns the covariance matrix of the source data with whitening and recolouring, which is applied to synthetic data in Sun and Saenko (2014).

Deep convolutional networks also had a dramatic impact in the field of domain adaptation. DeCAF (Donahue et al., 2014) demonstrated how features extracted from CNNs outperform by a large margin classification accuracies of commonly used features after adaptation, e.g., Bag of Words (Csurka et al., 2004) or HOG (Dalal and Triggs, 2005) features. While the standard adaptation techniques estimate the alignment after extracting the features, several papers opted for training deep networks by combining source and target datasets with specific architectures and loss functions that jointly minimised the classification regressor and the distance between domains (Ganin and Lempitsky, 2015; Ghifary et al., 2014; Long et al., 2016; Tzeng et al., 2015).

3. Adapted synthetic data for viewpoint refinement and estimation

In this section we describe the automatic process of refining coarse annotations of real data into fine viewpoints using adapted synthetic data. As depicted in Fig. 4, we initially request humans to coarsely annotate viewpoints of given 2D bounding boxes. Additionally, we also generate synthetic data with fine viewpoint annotations. This process is discussed in Section 3.1. Then, we adapt the synthetic data towards the real data, explained in Section 3.2, and assign fine viewpoints to the real data, further detailed in Section 3.3. We evaluate our approach for viewpoint refinement and viewpoint estimation. For viewpoint refinement, the coarse viewpoint is given and the goal is to estimate the fine viewpoint. For viewpoint estimation, the refined real and adapted synthetic data is used to train an estimator for fine-grained viewpoint estimation. The estimator is then evaluated on unseen test instances.

3.1. Generation of synthetic data from 3D models

In order to produce thousands of synthetic images, we first download free available 3D graphics models from the Internet. We then render the models, centred in the screen coordinate system, with 8 different light sources evenly spread around the object. Based on a Phong reflection model (Phong, 1975), we emphasise the usage of diffuse lighting to highlight shape variations and deformations, reducing the impact of ambient illuminations and specular reflections. The resulting rendered virtual classes used in the experiments are shown in Fig. 5(a). The scene is completed with a real background image taken from Geiger et al. (2012) placed behind the rendered object.

Finally, the generation process reduces to a parameterised camera

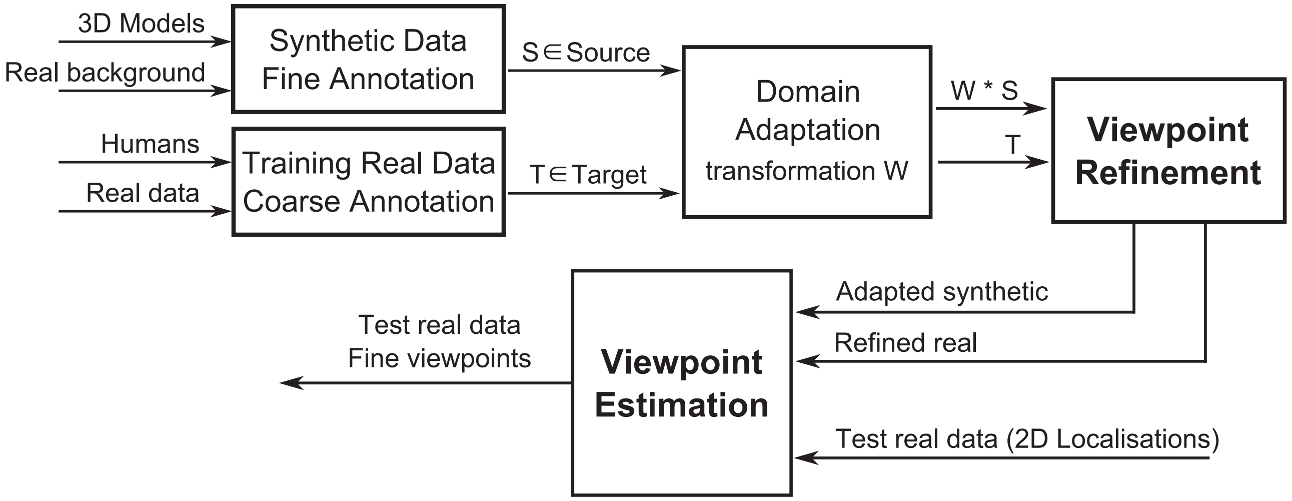


Fig. 4. Proposed pipeline for viewpoint refinement and estimation of real data.

displacement with azimuth θ, elevation ϕ and object distance r. Although this configuration allows to move along the whole view-sphere, we simplify the fine viewpoint annotations to the Y-axis rotation, being the azimuth angle the most dominant factor to recognise viewpoint differences in feature space, as well as the most relevant plane in viewpoint estimation tasks (He et al., 2014). Fig. 5(b) shows some examples of synthetic images. While the process of synthesizing images does not require much effort, it does not generate realistic images since the unknown 3D geometry and light conditions of the background are not taken into account.

3.2. Domain adaptation of synthetic data

Since synthetic data and real images belong to different domains, as illustrated in Fig. 3(b), we adapt the domain of the synthetic data to the real data. Our approach clusters the source (synthetic) and target (real) domains, and establishes correspondences between the clusters. The correspondences are then used to learn a mapping from the source domain to the target domain. The viewpoint annotations of the real images are then refined with viewpoint classifiers trained on the transformed synthetic data.

The learning of the mapping from the source to the target domain is discussed in Section 3.2.1 and the establishment of correspondences between clusters of both domains is discussed in Section 3.2.2.

3.2.1. Alignment from synthetic to real domain

To map the source data to the target domain, we have to learn a mapping from $\mathscr{S} \in \mathbb{R}^D$ to $\mathscr{T} \in \mathbb{R}^D$, where D denotes the dimensionality of the features. For label refinement, the dimensionality of the source and the target domain is the same. We consider a linear transformation, which is represented by a matrix $W \in \mathbb{R}^{D \times D}$, i.e., $t = Ws$.

Let $S = \{s_1, \ldots, s_M\}$ and $T = \{t_1, \ldots, t_N\}$, where $s \in S$ and $t \in T$, denote the training samples of the source and target domains, respectively. M and N are the total amount of samples of each domain and we can assume that $M \geq N$, since we can always generate more synthetic data than annotated real images. We first assume that for a subset of the target elements t_k we have already established a corresponding element in the source domain. The establishment of the correspondences $C = \{c_1, \ldots, c_K\}$ with (s_{c_k}, t_k) and $K \leq N$ will be explained in Section 3.2.2.

Given the correspondences, W can be learned by minimizing the objective

$$f(W) = \frac{1}{2}\sum_{k=1}^{K} \|Ws_{c_k} - t_k\|_2^2, \tag{1}$$

which can be expressed in matrix form:

$$f(W) = \frac{1}{2}\,\|WP_S - P_T\|_F^2. \tag{2}$$

The matrices P_S and $P_T \in \mathbb{R}^{D \times K}$ represent all assignments between source and target elements, where the columns denote the actual correspondences. We optimise the objective by non-linear optimisation. To this end, the derivatives of (2) are calculated by

$$\frac{\partial f(W)}{\partial W} = W(P_S P_S^T) - P_T P_S^T. \tag{3}$$

In our implementation, we use the local gradient-based optimization method of moving asymptotes (Svanberg, 2002), which is part of the NLOPT package (Johnson).

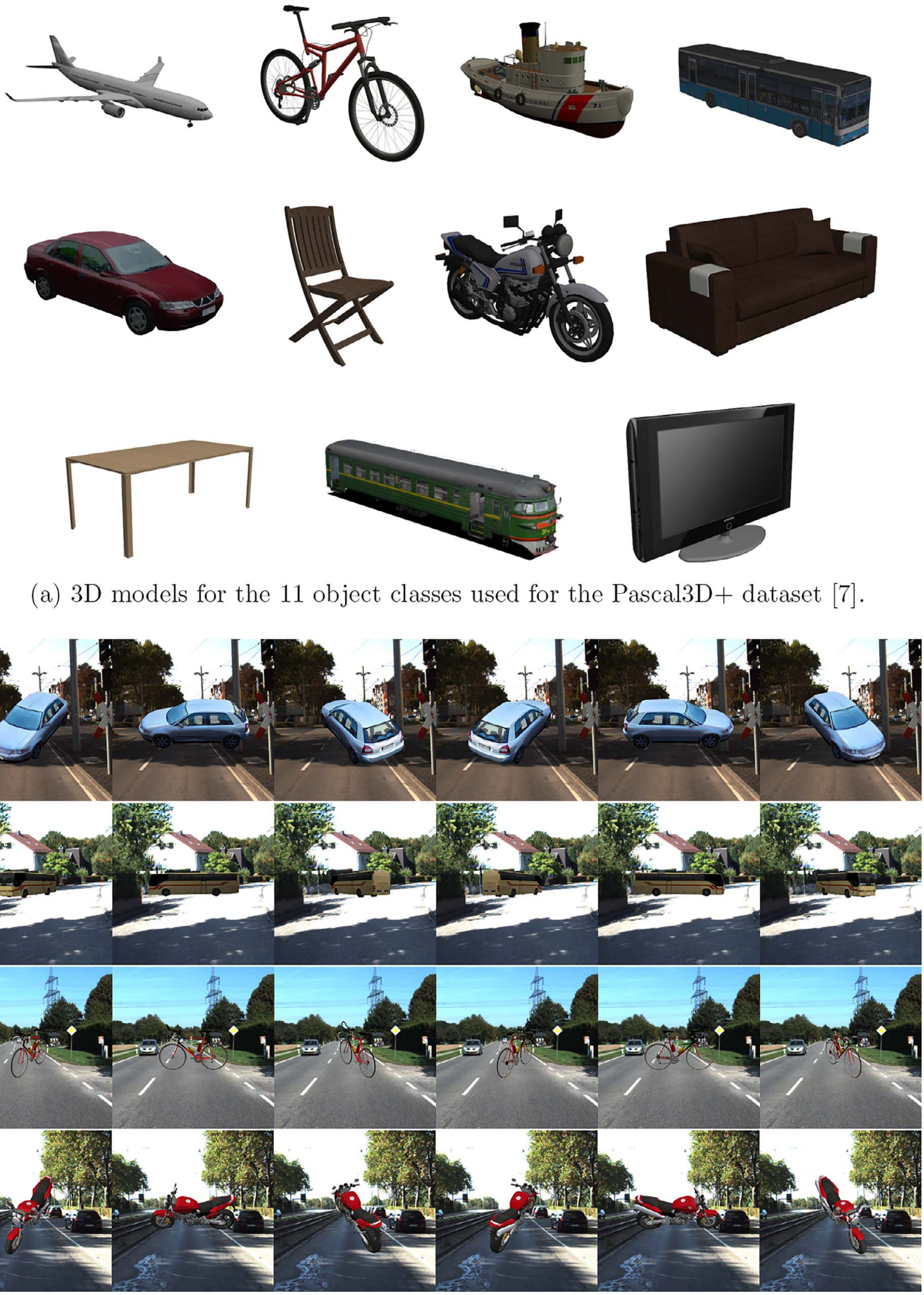
3.2.2. Source-target correspondences

In order to minimize (1), we first have to establish correspondences between the source and the target data. To this end, we cluster the data in both domains. For the synthetic data, we use the known fine-grained poses where each pose can be associated with one of the four coarse viewpoints i = {front, back, left, right}, i.e., $V = \sum_i V_i$, where V_i is the number of fine viewpoints for refinement in each coarse region. Fine viewpoints that lie between two coarse views are always assigned to the front or back views. For the target domain, we only have the coarse viewpoints and therefore cluster the N_i training samples of one coarse viewpoint further by K-Means, where the number of clusters for each coarse viewpoint is given by K_i, i.e., $K = \sum_i K_i$. and $V_i \leq K_i \leq N_i$. If $K_i = N_i$ clustering is not performed since each target instance is considered as one cluster. If $K_i = V_i$, the number of clusters is equal to the number of fine viewpoints. For the clustering, we represent each image by a HOG or CNN feature vector and append the aspect ratio of the bounding box surrounding the object.

As illustrated in Fig. 6, we establish correspondences between the clusters in the source and target domains, separately for each coarse viewpoint. To this end, we represent each cluster by its centroid. The sets of centroids are denoted by $\widehat{S}^i = \{\hat{s}_1^i, \ldots, \hat{s}_{V_i}^i\}$ and $\widehat{T}^i = \left\{\hat{t}_1^i, \ldots, \hat{t}_{K_i}^i\right\}$. The correspondences are then established by solving a bipartite matching problem:

$$\begin{aligned} &\operatorname*{argmin}_{e_{vk}} \sum_{v=1}^{V_i}\sum_{k=1}^{K_i} e_{vk}\left\|\hat{s}_v^i - \hat{t}_k^i\right\|_2^2 \\ &\text{subject to } \sum_v e_{vk} = 1 \quad \forall\, k\,, \quad \sum_k e_{vk} = a_v \quad \forall\, v \quad \text{and} \quad e_{vk} \\ &\in \{0, 1\} \quad \forall\, v, k. \end{aligned} \tag{4}$$

It assigns to each cluster in the target domain a unique cluster in the



(a) 3D models for the 11 object classes used for the Pascal3D+ dataset [7].

(b) Synthesised images with different azimuth, elevation and distance configurations.

Fig. 5. 3D graphics models for different object classes are rendered in front of real background images from Geiger et al. (2012) in order to automatically generate thousands of synthetic images with different accurate viewpoint annotations.

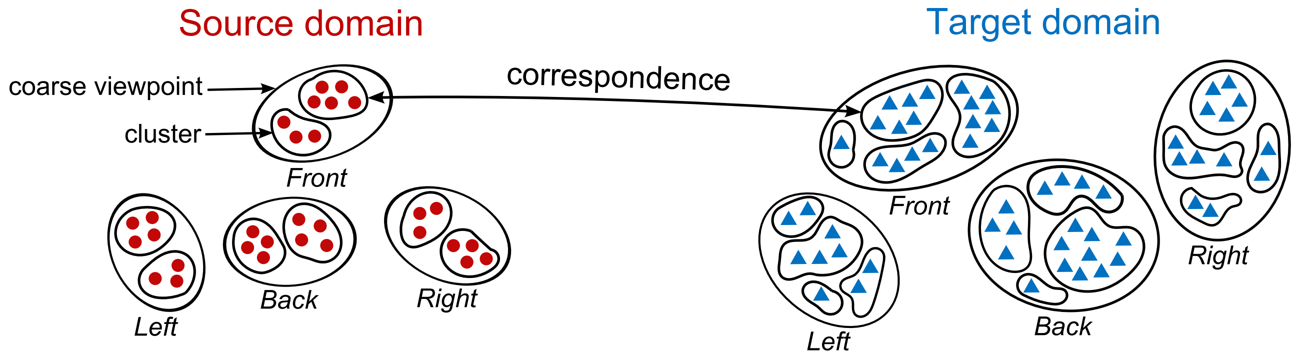


Fig. 6. Each cluster in the target domain is assigned to a source cluster that belongs to the same coarse viewpoint. In this example, for an 8-view refinement: $V_i = 2$ and $K_i = 4$.

source domain. Since there can be more clusters in the target domain than in the source domain, each source is associated to $a_v = K_i/V_i$ target clusters. If K_i is not a multiple of V_i, i.e., $aV_i < K_i < (a+1)V_i$, we set $a_v = a + 1$ for the first $K_i - aV_i$ source clusters and $a_v = a$ otherwise. We use the Hungarian algorithm (Kuhn, 1955) to solve the problem and for any cluster pair with $e_{vk} = 1$, we obtain a correspondence c_k. Due to the nature of the Hungarian method, which requires all combinations of centroid distances to be precomputed, we have not observed noticeable differences when solving Eq. (4) with other norms. The correspondences from all coarse views are then used to estimate the transformation W in Eq. (1).

3.3. Viewpoint refinement and estimation

The last step in our pipeline is the viewpoint refinement of the real training images. This is seen as a classification problem where we train on the transformed synthetic samples a linear SVM for each of the fine viewpoints $v = \{1, \ldots, V\}$, as effectively presented in other works (Glasner et al., 2011; Liebelt and Schmid, 2010; Pepik et al., 2012). Then, we apply the linear SVMs corresponding to the coarse viewpoint i of the real image and assign the fine pose with the highest scoring function:

$$f(x, i) = \underset{v=\{1,\ldots,V_i\}}{\operatorname{argmax}} \; w_v^T x + b_v, \tag{5}$$

where w_v and b_v are the weights and bias of the linear SVM for the fine viewpoint v. Since the transformation of synthetic data is guided by correspondences that deal with a discretised representation of viewpoints, i.e., source samples are clustered in exactly V centroids, we consider that the usage of classifiers for the final viewpoint refinement naturally fits in the overall formulation.

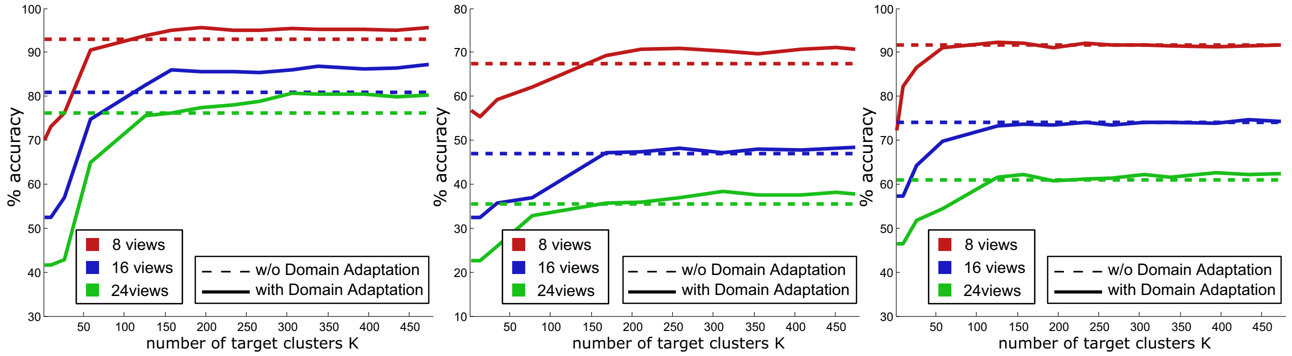
For pose estimation on real test images, we also use linear SVMs in a one-vs-all classification procedure. For each fine viewpoint, we train a linear SVM using the real training images with refined pose labels and the synthetic training images, which have been transformed by domain adaptation, together.

4. Experiments

We evaluate our algorithm on three car and three multi-object datasets with fine annotated poses. From the former group, the *Multi-View Car* (Ozuysal et al., 2009) dataset contains sequences of 20 cars as they rotate by 360°, where one image is taken every 3-4°. These fine-grained poses allow us to test the refinement at higher levels of viewpoint discretisation. We take the first 10 car sequences as training (1179 images) and the last 10 as test data (1120 images). Since the cars in this dataset are in a fixed location, we also evaluate our method on the more realistic *KITTI* (Geiger et al., 2012) benchmark, where images are recorded while driving along streets and roads. Due to the lack of bounding box annotations in the test data, we perform a 2-fold cross validation on the fully visible cars of the training set, containing 7481 images with 17,463 cars, 7811 of those which are non-occluded. For a cross-dataset experiment in Section 4.5, we also use the dataset (Sedaghat and Brox, 2015) where the bounding boxes and viewpoints have been annotated in a fully unsupervised manner. From the latter, the *3D Object Categorization* (Savarese and Fei-Fei, 2007) dataset provides 10 image sets of cars and bikes in 8 different angles (every 45°), permitting a refinement from 4 to 8 fine viewpoints. There are 2 elevations and 3 distances for each view, giving 48 images per object. We take 7 sets for training and 3 for testing. We also evaluate the method on the *Pascal3D+* (Xiang et al., 2014), which contains occlusions and truncated object instances of several classes. The main part of this dataset enriches the PASCAL VOC 2012 (Everingham et al., 2010) categories with 3D annotations for 11 rigid objects[1]: *aeroplane, bike, boat, bus, car, chair, dining table, motorbike, sofa, train* and *tv monitor*. The dataset has been further increased by images from the ImageNet dataset (Deng et al., 2009), which are also augmented with 3D annotations for the same rigid objects, and contain a larger amount of samples but with reduced number of occluded instances. Therefore, we opt for evaluating both subsets separately, denoted in our experiments as *Pascal3D* and *ImageNet3D*, respectively, using their validation sets as test data. The setup for the experiments is as follows. At first, we automatically generate synthetic data of textured 3D models for each object class. Following the evaluation protocol of Busto et al. (2015), we take 10 graphics models for each of the 11 rigid object categories, thus decreasing the number of cars from 15 to 10 in order to keep an even quantity among all classes. The attached background images, randomly taken from the KITTI dataset (Geiger et al., 2012), point towards the car's driving direction, allowing for synthetic vehicle placements, e.g., bike, bus, car and motorbike classes, in the centre of the image. In comparison to Busto et al. (2015), the synthetic images are obtained with a finer viewpoint granularity, rotating the θ angle of the camera every 1° in clockwise order, instead of every 10°, allowing for a total of 360 fine viewpoints. Since elevation ϕ varies among the objects classes, we take the elevation ranges of each object class from the training data of Xiang et al. (2014) and discretise them in 4 different levels, independently. Besides, we make use of one single distance, $r = 2.0$, in virtual world coordinates. The pose labels are then quantised to their closest angle of the V fine poses. The first viewpoint $v = 1$ lies at $\theta = 0$ in all quantisation levels. Overall, we generate 14,400 samples per object class. Some examples of the synthesised data are illustrated in Fig. 5(b).

Our first evaluation, in Section 4.1, measures the accuracy of our viewpoint refinement, extracting the bounding boxes of the real training images and converting the given viewpoints into the four coarse views, that is: *front* = (315°, ...,45°), *right* = [45°, ...,135°], *back* = (135°, ...,225°) and *left* = [225°, ...,315°]. Then, in Section 4.2, we evaluate the viewpoint estimation of the real test images having as

[1] In the standard protocol of Xiang et al. (2014), the class "bottle" is discarded due to its lack of viewpoint reference.



(a) Multi-View Car dataset (b) Pascal3D dataset (c) ImageNet3D dataset

Fig. 7. Impact of the number of target clusters *K* for viewpoint refinement.

training the adapted synthetic data and the refined real data. We use the given bounding boxes if the images are not already cropped. Neither coarse nor fine viewpoints are used for the test images. Section 4.3 discusses the impact of occluded object instances and Section 4.4 evaluates the accuracy of CNN-based methods for pose estimation using the refined datasets. We finally perform a cross-dataset experiment in Section 4.5.

Several widely used feature descriptors are evaluated to measure the performance of the method in different feature spaces. For the hand-crafted features, we rescale the bounding boxes to 128 × 128 pixels and extract HOG descriptors (Dalal and Triggs, 2005) with 8 bins (31 channels/bin), as in Busto et al. (2015). For the deep features, we take the AlexNet (Krizhevsky et al., 2012) and VGG (Simonyan and Zisserman, 2014) models and we extract the feature maps from the last convolutional layer (CNN-pool5), with 9216 and 25,088 dimensions from the standard 227 × 227 and 224 × 224 input patches, respectively. As we will show in Section 4.1, we reduce the dimensionality for AlexNet to 3041 dimensions (33%) and for VGG to 6272 dimensions (25%) without loss of accuracy. Additionally, we also evaluate the features from the last fully connected layer (CNN-fc7) of a re-trained VGG model, using the synthetic dataset and modifying the output layer with 360 classification channels. In the experiments with hand-crafted features, the annotated instances are rescaled preserving the aspect ratio. For the evaluations with deep features, the annotations are warped as in Tulsiani and Malik (2015).

4.1. *Viewpoint refinement*

We first evaluate the accuracy of our approach for pose refinement on the real training images. To this end, we use the coarse labels of the real training images and refine the viewpoints as described in Section 3.3. We then evaluate the accuracy of the refined labels on the real training images in conjunction with the transformed synthetic samples after the domain adaptation process. For the initial parameter evaluation of our technique, we stick to extracted AlexNet (CNN-pool5) features of car models. Then, we test the performance of our viewpoint refinement for all descriptors and classes.

4.1.1. *Impact of number of target clusters*

As described in Section 3.2.2, we cluster each coarse view by K-Means. We therefore evaluate the impact of the number of target clusters *K* on the viewpoint refinement. The results for the different datasets and *V* refined viewpoints used for evaluation are shown in Fig. 7. As baseline, we use linear SVMs trained on the synthetic data without domain adaptation. The accuracy tends to stabilize when the number of clusters is sufficiently large. The finer the viewpoints are the more clusters are also needed.

4.1.2. *Impact of number of target samples*

Although annotating real images by coarse viewpoints is easy to do, it also takes time. We therefore evaluate the impact of the number of coarsely labelled target samples *N*. To avoid any clustering artefacts, we set $K_i = N_i$, i.e., each target sample itself is a cluster. We also keep the numbers of the real images N_i for each of the four viewpoints equal while increasing *N*. The results in Fig. 8 show that already 100–150 annotated samples per coarse view give a boost in performance compared to the baseline. This means that very little time is actually required for the annotation task.

4.1.3. *Impact of number of 3D models*

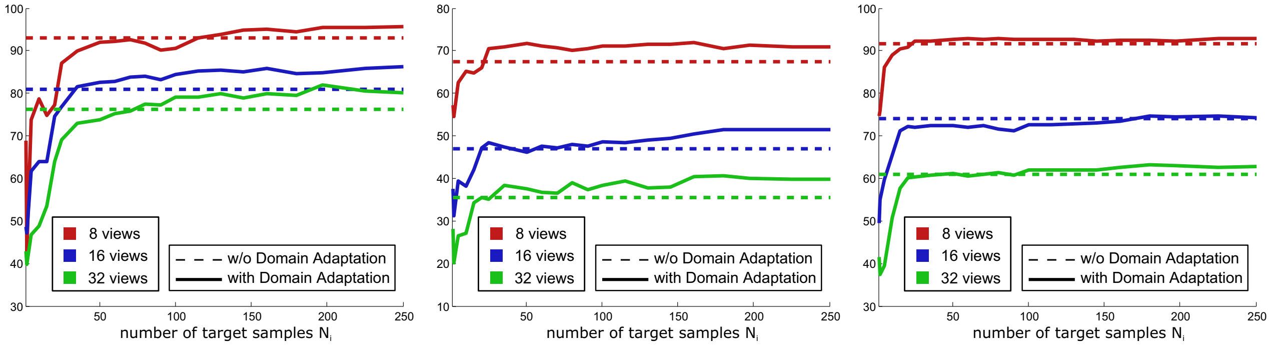
We also evaluate the impact of the amount of 3D models used to generate synthetic data. Fig. 9 shows how the accuracy tends to stabilise with already 5 models.

4.1.4. *Weak supervision*

If the target samples are not annotated by the four coarse views, we can still perform unsupervised domain adaptation. In this case, we observe a substantial amount of wrong viewpoint estimates by 180° as shown by the confusion matrix in Fig. 10(a). In contrast, we resolve these errors by using the coarse viewpoints of the real images as weak supervision as shown in Fig. 10(b). This shows that using coarse annotations of real images, which are inexpensive to annotate, significantly increases the viewpoint refinement accuracy.

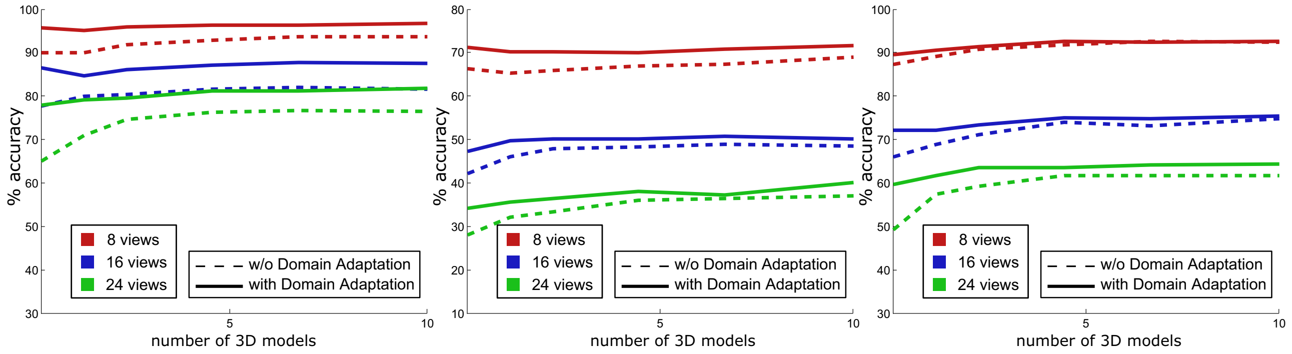
4.1.5. *Accuracy of the viewpoint refinement*

We finally compare the refinement accuracy of our method with popular domain adaptation techniques (Fernando et al., 2013; Gong et al., 2012; Sun et al., 2015). The geodesic flow kernel (GFK) (Gong et al., 2012) is an unsupervised domain adaptation method that maps both domains to a common subspace in a Grassmannian manifold. The same applies to the sub-space alignment technique (SA) (Fernando et al., 2013), that maps both domains to a common subspace using the *d* largest eigenvectors. In both cases, the number of chosen sub-dimensions *d* is kept as large as possible to avoid a significant loss in accuracy. Lastly, we also test the current state-of-the-art adaptation method named CORAL (Sun et al., 2015). Without any dimensionality reduction, it decorrelates the source samples by whitening and re-colours them by the covariance matrix of the target data. For all methods, we exploit the weak supervision and apply them for each coarse viewpoint, independently. As already shown in Busto et al. (2015), supervised methods that internally process the coarse labelling (Hoffman et al., 2013) report worse viewpoint accuracies than the unsupervised methods. For the refinement after domain adaptation, we use linear SVMs as described in Section 3.3. As baseline, we use the linear SVMs trained on the synthetic data without domain adaptation (*w/o DA*).



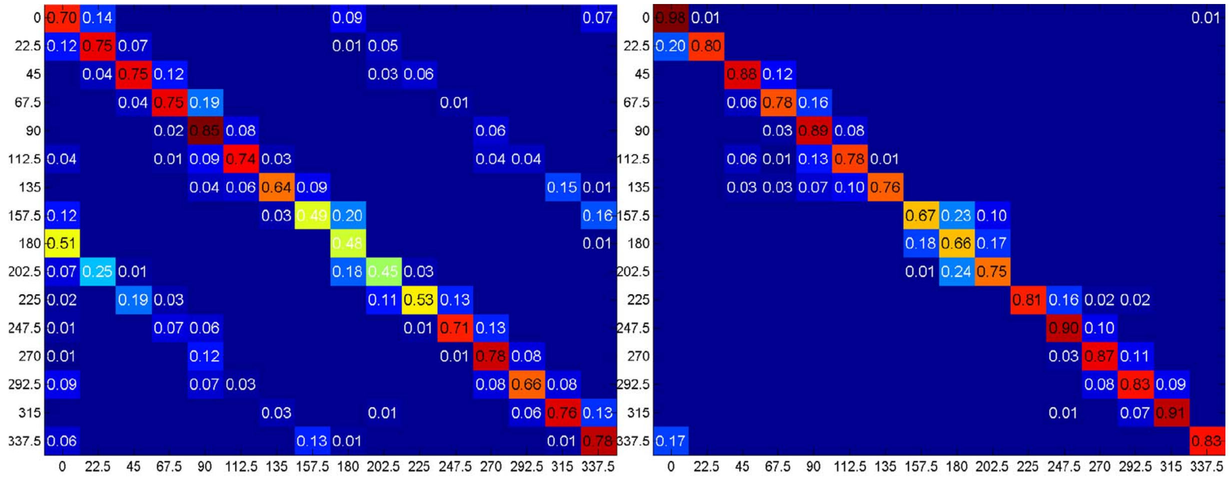
(a) Multi-View Car dataset (b) Pascal3D dataset (c) ImageNet3D dataset

Fig. 8. Impact of the number of target samples N_i per coarse view for the refinement.



(a) Multi-View Car dataset (b) Pascal3D dataset (c) ImageNet3D dataset

Fig. 9. Impact of the number of 3D car models for viewpoint refinement.



(a) Unlabelled target samples (b) 4 viewpoint labels in target samples

Fig. 10. Confusion matrix for the Multi-View Car dataset in a 16-viewpoint refinement. (a) Without supervision rotations by 180° are sometimes confused. (b) When weak supervision from the four coarse viewpoint labels is used, these confusions are resolved.

For our method, we report the refinement accuracy for four different clustering settings. For the first three, we set V equal to the number of views for fine-grained viewpoint estimation as in the previous experiments. We report numbers for $K = V$, $K = 100$ and $K = N$. For the first two settings, we report the mean accuracy and its standard deviation over 10 runs since K-Means depends on the random initialization. In the last setting, each target sample is a cluster.

We first report the results only for the fully visible object exemplars and compare the hand-crafted features (HOG) and the deep features, i.e., the last convolutional features from AlexNet and VGG models (CNN-pool5) after dimensionality reduction and the re-trained fully connected layer of VGG (CNN-fc7), in Table 1. The accuracies of CNN-pool5 features from both models outperform the results of the HOG features, obtaining VGG slightly better results than AlexNet, especially for finer viewpoints. While both CNN-pool5 features achieve the best overall results, VGG CNN-fc7 performs slightly better on the Multi-View Car dataset for $V \geq 72$.

While $K = N$ performs best in almost all cases, $K = 100$ and $K = V$ achieves the highest accuracy in only very few cases, with only marginal improvements compared to $K = N$. Overall, $K = N$ with CNN-

Table 1
Accuracy of the coarse-to-fine viewpoint refinement for different domain adaptation techniques. For the methods with K-Means clustering, the mean and standard deviation (brackets) over 10 runs are provided.

Views	3DObjCat (Savarese and Fei-Fei, 2007)		Multi-view car (Ozuysal et al., 2009)							KITTI (Geiger et al., 2012)	
	HOG										
	8/car	8/bike	8	16	24	36	72	180	360	8	16
w/o DA	98.2	98.4	88.8	78.1	68.5	55.6	30.9	13.3	6.5	82.5	**69.9**
GFK (Gong et al., 2012)	98.2	98.0	89.3	79.7	71.3	55.9	31.7	14.9	6.5	82.2	69.1
SA (Fernando et al., 2013)	96.4	98.4	87.5	77.0	69.4	55.2	31.8	13.2	6.4	**87.4**	69.3
CORAL (Sun et al., 2015)	95.8	95.8	89.9	79.1	65.7	52.4	24.6	10.2	4.4	78.3	66.5
V = views, K = V	83.4	81.6	78.6	63.7	63.0	51.5	30.6	14.5	7.0	65.7	65.8
	(0.8)	(0.7)	(1.7)	(2.1)	(2.2)	(1.8)	(1.4)	(1.0)	(0.6)	(1.9)	(1.4)
V = views, K = 100	99.4	98.8	**92.1**	81.2	71.1	59.3	32.2	14.6	**7.6**	80.4	67.5
	(0.2)	(0.4)	**(0.6)**	(0.8)	(1.5)	(1.2)	(1.1)	(0.9)	**(0.4)**	**(1.4)**	(1.5)
V= views, K = N	**100.0**	**99.8**	91.0	**85.3**	**76.8**	**64.4**	**38.1**	**15.6**	7.4	83.8	69.0
V = M, K = N	98.2	98.4	89.3	78.1	67.6	53.8	28.8	13.4	7.1	82.0	67.5
	AlexNet CNN-pool5										
w/o DA	99.7	97.0	93.5	81.5	76.8	61.4	35.1	12.8	6.1	81.9	70.4
GFK (Gong et al., 2012)	99.4	97.8	94.9	83.5	78.5	60.4	35.1	14.4	6.8	83.2	67.4
SA (Fernando et al., 2013)	99.7	96.8	92.5	81.4	76.1	61.1	35.5	13.0	6.8	83.5	**71.3**
CORAL (Sun et al., 2015)	98.8	94.8	94.4	81.5	71.5	54.8	27.1	7.0	2.0	79.7	64.1
V = views, K = V	83.3	68.5	70.8	52.7	42.2	29.2	30.2	14.4	8.3	67.3	40.1
	(1.7)	(2.2)	(2.7)	(1.5)	(1.3)	(1.7)	(1.0)	(0.5)	(0.8)	(2.2)	(2.8)
V = views, K = 100	99.7	95.6	94.7	83.2	71.2	56.4	30.9	14.5	**8.7**	75.9	64.9
	(0.0)	(0.9)	(0.5)	(1.2)	(1.1)	(1.4)	(1.2)	(0.8)	**(0.8)**	(1.9)	(1.7)
V = views, K = N	**100.0**	**99.0**	**96.7**	**87.5**	**81.7**	**67.7**	**40.5**	**16.3**	7.3	**84.7**	68.8
V = M, K = N	99.7	97.0	93.6	81.2	71.4	60.0	34.3	13.3	6.9	82.1	63.3
	VGG CNN-pool5										
w/o DA	99.7	96.2	93.5	84.4	76.2	62.5	34.5	13.0	6.7	82.1	68.3
GFK (Gong et al., 2012)	99.4	97.0	95.1	85.0	78.1	61.0	33.9	14.0	7.1	83.1	66.1
SA (Fernando et al., 2013)	98.2	91.3	93.3	83.9	75.5	59.6	33.4	13.2	7.2	82.5	67.6
CORAL (Sun et al., 2015)	98.2	94.6	95.0	82.8	75.4	60.0	31.3	9.8	4.6	77.2	65.8
V = views, K = V	54.5	60.1	54.8	37.0	22.4	25.4	20.4	11.9	7.9	49.5	30.7
	(3.4)	(3.7)	(4.1)	(3.2)	(2.8)	(2.0)	(1.7)	(1.0)	(1.1)	(4.0)	(2.2)
V = views, K = 100	97.3	93.5	92.6	73.6	60.2	41.1	21.2	12.4	8.0	82.0	64.5
	(0.5)	(0.6)	(0.7)	(1.1)	(1.3)	(0.9)	(0.5)	(0.8)	(0.6)	(1.0)	(1.3)
V = views, K = N	**100.0**	**98.8**	**95.5**	**87.0**	**82.1**	**70.1**	**42.7**	**19.5**	**9.0**	**84.7**	**68.5**
V = M, K = N	99.4	96.2	93.6	84.1	72.2	60.8	34.0	13.3	7.5	82.5	62.2
	VGG CNN-fc7										
w/o DA	96.4	96.8	90.6	79.8	74.2	63.6	43.2	20.2	9.0	78.9	65.3
GFK (Gong et al. (2012))	96.5	96.9	91.2	81.1	76.0	63.3	42.8	19.9	10.0	78.1	64.5
SA (Fernando et al. (2013))	95.5	96.2	90.5	80.1	73.4	61.9	**43.6**	18.9	10.7	79.4	64.7
CORAL (Sun et al. (2015))	91.7	94.1	**93.9**	**83.6**	76.0	**63.9**	42.0	19.1	9.2	74.9	59.6
V=views, K=V	86.9	**97.4**	89.8	72.9	69.3	58.9	40.9	19.7	**10.4**	66.4	56.3
	(1.2)	**(0.5)**	(1.0)	(1.5)	(2.0)	(1.8)	(2.1)	(0.9)	**(0.7)**	(2.3)	(1.7)
V = views, K = 100	97.0	96.8	90.9	80.2	**76.2**	63.8	43.5	21.5	10.2	76.8	63.2
	(0.5)	(0.5)	(0.7)	(0.9)	**(0.9)**	(1.1)	(0.8)	(0.8)	(0.6)	(2.1)	(2.2)
V = views, K = N	**97.9**	96.8	90.8	80.6	74.8	**63.9**	43.3	**22.2**	9.9	78.6	**67.2**
V = M, K = N	96.4	97.0	90.7	81.4	75.9	63.6	39.1	20.2	9.7	**79.8**	64.4

pool5 features performs best.

We also evaluated the accuracy when V is also set to the number of synthetic samples M, i.e., each synthetic image is a cluster. In this case, the accuracy drops significantly for all datasets and feature descriptors. This shows that the synthetic data needs to be quantized according to the fine-grained views.

Table 1 also compares our approach to other domain adaptation methods (Fernando et al., 2013; Gong et al., 2012; Sun et al., 2015). In nearly all setting and feature combinations, our method outperforms the generic domain adaptation methods. Although CORAL obtains better results with CNN-fc7 features, the reported accuracies are still lower than the results of the CNN-pool5 features with our method.

In contrast to the datasets (Geiger et al., 2012; Ozuysal et al., 2009; Savarese and Fei-Fei, 2007), the datasets Pascal3D and ImageNet3D contain many occluded and truncated objects. The results for these two datasets are reported in Table 2. We report the accuracies for both CNN models with the CNN-pool5 features using $K = N$ and compare it to the baseline without domain adaptation. Except for the 8 view refinement on ImageNet3D, our approach outperforms the baseline by around 4–6%. In general, the reported results of the AlexNet and VGG models are comparable.

4.1.6. *Impact of dimensionality reduction*

For the results shown in Tables 1 and 2, we reduced the dimensionality of the convolutional feature maps. Since in most of the experiments $D > M + N$, we employ randomised singular value decomposition to reduce the dimensionality for efficiency. Fig. 11 shows that deep features from convolutional layers can be strongly reduced. While the performance of the HOG features start to decrease with less than 40% of the feature dimensionality, the dimensionality of the AlexNet and VGG CNN-pool5 features can be reduced without significant loss in accuracy by 33% and 25%, respectively.

Table 2
Accuracy of the coarse-to-fine viewpoint refinement for the Pascal3D and ImageNet3D datasets that contain occlusions and truncated object instances.

		PASCAL3D (Xiang et al., 2014)											
		AlexNet CNN-pool5											
Views		Aero	Bike	Boat	Bus	Car	Chair	Table	mbike	Sofa	Train	TV	Avg.
8	w/o DA	**63.4**	67.5	57.6	69.3	68.2	58.6	64.9	**70.6**	61.4	65.2	64.4	64.6
	V = views, K = N	59.0	**68.6**	**60.7**	**72.0**	**70.9**	**63.2**	**66.5**	70.1	**66.0**	**67.8**	**68.4**	**66.7**
16	w/o DA	**42.0**	42.0	31.5	53.9	47.8	38.4	41.3	44.8	43.2	42.6	**41.2**	42.6
	V = views, K = N	36.0	**49.3**	**35.1**	**57.6**	**49.5**	**42.4**	**45.0**	**54.5**	**44.1**	**47.4**	34.3	**45.0**
24	w/o DA	**28.6**	34.2	19.2	43.8	36.2	29.4	**28.7**	34.1	32.3	23.7	19.7	30.0
	V = views, K = N	28.0	**39.8**	**27.5**	**44.7**	**39.3**	**31.1**	27.7	**39.4**	**35.7**	**30.4**	**35.6**	**34.5**
		VGG CNN-pool5											
8	w/o DA	**62.1**	67.3	55.6	**68.7**	67.8	60.0	**47.9**	68.5	64.1	66.8	55.4	62.2
	V = views, K = N	57.4	**72.5**	**58.2**	67.9	**70.7**	**63.6**	46.7	**69.4**	**76.0**	**70.1**	**59.7**	**64.7**
16	w/o DA	**38.7**	40.9	32.1	**62.7**	46.4	36.1	34.6	47.5	35.3	38.7	43.8	41.5
	V = views, K = N	36.1	**53.2**	**36.0**	56.8	**50.7**	**41.5**	**45.2**	**52.1**	**40.0**	**52.4**	**49.1**	**46.6**
24	w/o DA	24.3	30.7	18.2	43.3	36.1	26.2	22.2	32.7	25.6	30.0	29.6	29.0
	V = views, K = N	**29.0**	**39.4**	**26.5**	**46.1**	**40.4**	**30.8**	**27.7**	**38.9**	**38.2**	**37.8**	**37.4**	**35.7**
		ImageNet3D (Xiang et al., 2014)											
		AlexNet CNN-pool5											
8	w/o DA	**64.8**	**78.8**	**56.5**	**94.9**	91.3	75.5	73.0	73.8	**77.4**	**64.8**	**81.6**	**75.7**
	V = views, K = N	60.1	78.7	55.9	92.8	**91.5**	**75.8**	**76.6**	**77.5**	77.2	63.6	**81.6**	75.6
16	w/o DA	**46.5**	56.0	36.3	70.7	73.6	62.1	34.9	52.1	57.0	34.7	**39.3**	51.2
	V = views, K = N	42.1	**60.0**	**37.8**	**74.6**	**74.2**	**62.5**	**60.0**	**58.8**	**63.8**	**45.5**	37.3	**56.1**
24	w/o DA	**37.5**	41.3	25.7	54.4	60.5	48.9	28.7	36.1	**45.0**	28.1	**40.0**	40.6
	V = views, K = N	36.8	**48.2**	**27.8**	**62.4**	**63.2**	**53.3**	**50.8**	**40.6**	43.4	**34.7**	35.3	**45.1**
		VGG CNN-pool5											
8	w/o DA	**64.8**	76.4	**60.7**	**92.2**	**91.5**	**77.3**	71.4	**71.8**	**85.1**	**77.4**	**80.5**	**77.2**
	V = views, K = N	61.1	**76.5**	58.3	87.9	90.1	73.7	**74.8**	77.7	73.8	70.7	74.5	74.5
16	w/o DA	**44.2**	55.0	36.7	69.0	**73.5**	55.8	44.1	52.5	57.6	**44.0**	25.2	50.7
	V = views, K = N	**44.2**	**59.1**	**38.6**	**73.3**	72.6	**59.0**	**57.9**	**57.1**	**60.3**	40.8	**46.5**	**55.4**
24	w/o DA	33.5	40.4	26.0	53.9	**63.5**	44.1	33.2	34.2	42.0	22.1	22.1	37.7
	V = views, K = N	**35.2**	**50.1**	**30.2**	**57.1**	63.2	**47.4**	**44.5**	**43.1**	**56.1**	**26.7**	**22.5**	**43.3**

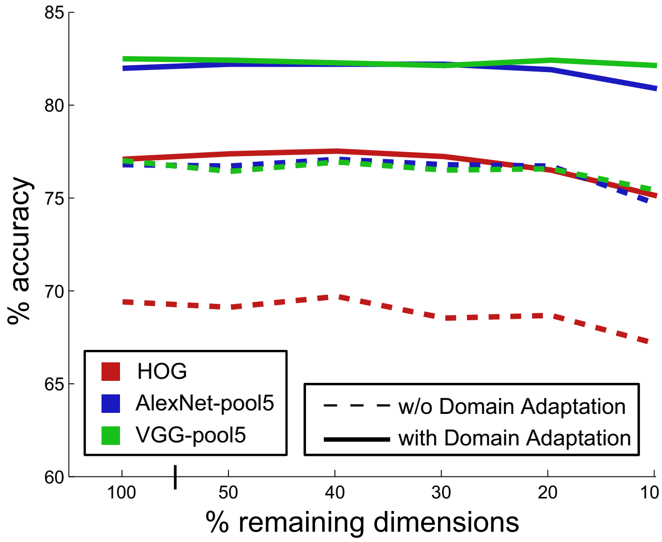


Fig. 11. Impact of dimensionality reduction using randomised singular value decomposition for different feature descriptors on the Multi-View Car dataset with a 24-viewpoint refinement setting.

4.2. Viewpoint estimation

We then evaluate the accuracy of the pose estimation on the real test images. To this end, we train the viewpoint estimator described in Section 3.3 on the synthetic data (*syn*), the real training data (*real*) with refined viewpoint labels or on both datasets (*joint*). For the refinement, we use our approach with $K = N$ (*with DA*) and compare it to the refinement without domain adaptation (*w/o DA*). We report the results for the datasets with non-occluded object instances in Table 3, where we also compare the accuracy of the pose estimator when the fine ground-truth viewpoint annotations of the real training images (*gt*) are used for training. This serves as an upper bound of the accuracy in comparison to the setting with only weak supervision.

When comparing the results of the domain adaptation for the synthetic, real or both training sets with the results without domain adaptation, we observe that the domain adaptation improves the viewpoint estimation for all scenarios with HOG and CNN-pool5 features, with the exception of the KITTI dataset with 16 viewpoint refinement, since it mainly contains cars facing coarse directions. On the contrary, the CNN-fc7 features only obtain minor improvements for some of the settings, which is consistent with the previous results.

Using refined real target images (*with DA real*) for training is in most cases sufficient. The adapted synthesized training data, however, performs better for fine-grained viewpoints $V \geq 72$ since the real images do not necessary provide enough samples for each viewpoint. Combining the real and synthetic data for training (*with DA joint*) also works very well for any viewpoint discretisation.

Table 4 reports the accuracies for the Pascal3D and ImageNet3D datasets using CNN-pool5 features from the VGG model. On these datasets the adapted synthesized training data performs already better than the real data for $V \geq 16$ fine viewpoints. As before, combining the refined real data and the adapted synthesized data for training performs well for any viewpoint discretisation $V = 8, 16, 24$. It is interesting to note that our weakly supervised approach (*with DA joint*) even outperforms the fully supervised approach (*gt*) due to the training data augmentation by the adapted synthetic images.

Table 3
Pose estimation accuracy on unlabelled test data using real training data, synthetic data or both training sets. All datasets contain non-occluded object instances.

		3DObjCat (Savarese and Fei-Fei, 2007)		Multi-View Car (Ozuysal et al., 2009)							KITTI (Geiger et al., 2012)	
		HOG										
	views	8/car	8/bike	8	16	24	36	72	180	360	8	16
	gt	*100.0*	*100.0*	*77.7*	*69.1*	*61.1*	*53.3*	*35.0*	*13.1*	*1.7*	*85.8*	*82.5*
w/o DA	syn	77.1	84.7	67.8	61.1	53.4	35.9	19.8	6.9	4.2	58.8	54.8
	real	**100.0**	99.1	76.2	64.5	53.5	41.3	20.8	2.7	0.6	77.4	64.8
	joint	87.5	97.7	74.1	65.7	55.5	43.7	22.5	6.5	4.3	**80.1**	**66.8**
With DA	syn	86.1	94.0	73.1	66.3	59.5	43.7	22.6	**8.5**	4.5	68.1	46.3
	real	**100.0**	**99.5**	**76.5**	**68.1**	**63.1**	**48.5**	22.8	7.8	1.1	76.9	61.7
	joint	91.0	98.2	74.2	67.9	62.0	45.4	**23.3**	8.0	**4.9**	77.8	62.9
		AlexNet CNN-pool5										
	gt	*100.0*	*98.2*	*82.6*	*74.8*	*68.1*	*57.1*	*33.5*	*12.0*	*1.7*	*92.5*	*85.0*
w/o DA	syn	91.0	81.0	84.5	66.8	55.8	44.5	23.7	8.5	4.5	60.3	46.2
	real	**100.0**	97.2	81.4	71.5	62.5	46.9	24.4	2.2	0.1	75.7	64.0
	joint	96.5	93.1	80.6	70.5	62.8	47.5	25.7	**10.6**	5.1	77.8	**66.4**
With DA	syn	98.6	95.4	82.4	73.2	61.4	50.2	26.7	9.3	**5.2**	69.5	36.0
	real	**100.0**	**97.7**	82.9	73.4	**66.3**	**53.5**	26.7	7.0	0.5	78.0	61.4
	joint	98.6	94.9	**83.0**	**74.8**	64.1	52.4	**27.3**	10.4	4.8	**78.5**	62.8
		VGG CNN-pool5										
	gt	*100.0*	*99.1*	*85.0*	*75.7*	*70.0*	*56.5*	*34.2*	*10.1*	*1.0*	*87.6*	*82.0*
w/o DA	syn	91.7	83.3	77.3	64.9	53.6	40.9	18.6	5.3	2.9	63.6	42.2
	real	**100.0**	97.2	83.3	73.3	65.2	45.4	20.8	2.8	1.2	75.4	63.7
	joint	97.9	95.8	81.9	73.3	61.9	48.5	20.9	7.2	2.8	79.0	**66.8**
With DA	syn	**100.0**	97.2	82.8	74.1	62.8	49.1	22.6	8.6	3.2	76.9	39.8
	real	**100.0**	**99.5**	**84.5**	74.5	**69.5**	**53.6**	**27.7**	**10.5**	1.6	77.1	61.7
	joint	**100.0**	98.6	83.9	**74.9**	63.9	50.9	23.6	10.1	**3.4**	**81.5**	62.9
		VGG CNN-fc7										
	gt	*88.2*	*93.1*	*71.2*	*66.0*	*60.6*	*51.2*	*34.2*	*14.7*	*0.8*	*81.5*	*71.1*
w/o DA	syn	76.4	78.2	67.3	61.2	55.0	44.7	26.0	**11.9**	4.5	59.8	49.6
	real	84.7	**91.7**	69.0	**62.0**	55.0	45.9	25.2	**11.9**	0.2	**71.8**	57.3
	joint	84.7	87.5	68.6	**62.0**	54.3	45.0	26.2	9.5	**6.1**	70.6	58.1
With DA	syn	79.9	82.9	67.4	61.0	55.6	44.9	**26.8**	10.2	5.9	62.6	49.0
	real	**87.5**	91.2	68.5	61.8	**55.8**	**46.0**	26.0	10.6	0.4	71.1	57.3
	joint	**87.5**	88.0	**69.6**	61.6	53.9	44.6	25.9	9.5	5.6	70.4	**58.5**

4.3. Occlusion

In order to measure the actual impact of occluded instances, we also compare the viewpoint refinement for the Pascal3D and ImageNet3D datasets when we only take non-occluded object instances for training and testing (*non-occ*). As shown in Table 5, the accuracies for the setting with non-occluded instances in comparison to the complete dataset (*all*) are higher as expected. This is especially the case for Pascal3D since it contains a smaller portion of fully visible samples, i.e., 38% vs. 75%. The gain of our approach compared to the baseline, however, remains similar for *all* and *non-occ* with +6.7% and +5.0%, respectively. This shows that our approach is robust to occlusions.

For completeness, we also evaluate the scenario for viewpoint estimation. Table 6 reports the accuracies of all four combinations depending if the training or test data contain occluded and truncated objects (*all*) or only fully visible objects (*non-occ*). For Pascal3D, the best average accuracies are obtained if occluded and truncated objects are discarded from the training data although the impact varies strongly among the object categories. For ImageNet3D, which contains by far less occluded samples, the best accuracy is achieved by taking all training samples. A major gain can be observed for the categories *bike, motorbike*, and *sofa*, which are the categories with the highest ratio of occluded or truncated samples.

4.4. Viewpoint estimation using CNNs

In order to demonstrate that our approach not only works with linear SVMs but also with other methods for viewpoint estimation, we use our approach to train a state-of-the-art CNN approach for viewpoint estimation (Tulsiani and Malik, 2015), which also models viewpoint estimation as a classification task. In addition, we modify the CNN for viewpoint regression by using Huber loss H of the azimuth angle θ in a continuous representation $F(\theta) = [cos(\theta), sin(\theta)]$ as in Massa et al. (2016). We augment the training data with mirrored samples and jittered ground-truth bounding boxes that overlap with the annotated bounding box with IoU $>$ 0.7. We run a total of 40,000 iterations for the CNNs trained with only real data and 60,000 for those that include the synthetic data. In both cases, we start with a learning rate of 0.001 and decrease it by a factor of 10 each time a third of the iterations are completed.

The results for the Pascal3D dataset are given in Table 7 where we report the viewpoint estimation accuracy for 24 views as in the previous tables and the median error (*MedError*) as it was used in Tulsiani and Malik (2015). When we train the CNN with classification loss on the training data with ground-truth labels, we achieve a lower median error and higher accuracy compared to the regression loss. This was already observed in Massa et al. (2016).

When the CNN is trained not on the ground-truth but on the refined viewpoint labels, our proposed approach with domain adaptation (*with DA*) outperforms the baseline (*w/o DA*) for all settings. Training on the synthetic and refined real training images (*joint*) also improves the accuracy and reduces the error compared to using the real training images only (*real*). We finally compare the CNN-based viewpoint classification (Tulsiani and Malik, 2015) with the linear SVMs (*DA LSVM*), which have been previously used for viewpoint estimation in Table 4. Using (Tulsiani and Malik, 2015) instead of linear SVMs

Table 4
Pose estimation accuracy for the Pascal3D and ImageNet3D datasets that contain occlusions and truncated object instances.

			PASCAL3D (Xiang et al., 2014)											
			VGG CNN-pool5											
	Views		Aero	Bike	Boat	Bus	Car	Chair	Table	mbike	Sofa	Train	TV	Avg.
8	w/o DA	*gt*	*49.0*	*46.0*	*29.4*	*36.6*	*43.2*	*44.4*	*25.0*	*52.8*	*26.4*	*21.6*	*18.6*	*35.7*
		syn	**40.7**	47.9	22.7	42.0	37.2	36.1	**22.9**	48.4	37.4	**32.3**	31.3	36.3
	with DA	syn	34.6	51.7	18.9	45.5	41.5	41.1	16.7	51.7	43.3	31.6	**35.7**	37.5
		real	34.8	**52.1**	**26.8**	33.8	41.6	42.1	18.0	**56.4**	27.1	18.4	24.4	34.1
		joint	35.3	50.9	19.6	**47.8**	**43.9**	**42.3**	17.7	51.4	**46.2**	30.7	34.4	**38.2**
16	w/o DA	*gt*	*29.7*	*23.6*	*16.0*	*21.7*	*28.5*	*25.7*	*12.4*	*28.7*	*18.8*	*15.5*	*10.3*	*21.0*
		syn	**22.2**	23.9	11.9	**30.4**	25.6	22.1	10.6	28.5	24.0	23.3	15.9	21.7
	with DA	syn	20.9	25.1	**13.0**	28.8	27.5	**26.3**	10.1	30.5	**25.5**	20.9	18.4	22.5
		real	21.3	23.8	12.0	22.8	28.2	22.6	12.4	30.6	17.4	17.2	12.8	20.1
		joint	21.4	**26.4**	11.8	29.4	**29.3**	25.1	**16.6**	**31.2**	25.0	**23.9**	**25.7**	**24.2**
24	w/o DA	*gt*	*23.1*	*16.1*	*10.7*	*17.9*	*21.9*	*18.9*	*7.7*	*16.1*	*12.8*	*13.6*	*11.0*	*15.4*
		syn	14.8	18.0	7.8	21.0	18.6	15.6	**8.0**	20.0	18.0	14.9	9.9	15.1
	with DA	syn	16.0	**18.9**	8.0	22.9	19.8	16.7	7.7	20.8	18.2	**16.6**	**14.4**	16.4
		real	15.6	16.7	8.3	21.0	**21.5**	15.1	7.6	**21.2**	13.6	12.1	8.2	14.6
		joint	**18.4**	18.6	**8.7**	**24.1**	20.7	**17.0**	7.6	**21.2**	**18.3**	15.4	14.2	**16.7**
			ImageNet3D (Xiang et al., 2014)											
			VGG CNN-pool5											
8	w/o DA	*gt*	*59.2*	*66.4*	*55.4*	*51.4*	*87.6*	*42.9*	*38.8*	*66.6*	*31.7*	*22.9*	*33.8*	*50.6*
		syn	40.3	62.9	27.2	65.7	75.4	60.4	2.1	60.1	46.6	30.0	22.4	46.6
	with DA	syn	40.8	69.3	35.8	72.7	80.8	59.7	39.0	**64.1**	60.8	27.5	47.5	54.4
		real	**43.3**	67.4	**40.7**	58.4	**84.6**	48.8	33.3	64.0	28.5	21.0	39.2	48.1
		joint	43.1	**71.4**	39.1	**77.1**	83.6	**61.1**	**44.6**	63.7	**61.4**	**36.8**	**48.6**	**57.3**
16	w/o DA	*gt*	*42.3*	*44.4*	*39.0*	*35.6*	*71.2*	*29.4*	*24.1*	*41.0*	*22.3*	*22.8*	*16.2*	*35.3*
		syn	**30.1**	35.5	13.0	46.8	60.6	37.7	12.3	35.8	**36.3**	15.1	10.3	30.3
	with DA	syn	26.4	**47.3**	20.4	52.8	66.4	33.0	23.5	42.3	36.1	25.1	31.3	36.8
		real	24.5	42.8	**23.9**	39.3	**68.2**	24.3	16.5	34.8	26.2	16.7	18.2	30.5
		joint	29.0	46.1	23.3	**54.8**	67.5	**42.1**	**23.8**	**44.1**	35.0	**27.7**	**34.7**	**38.9**
24	w/o DA	*gt*	*28.5*	*32.1*	*30.8*	*31.7*	*62.6*	*2.7*	*18.9*	*28.1*	*13.6*	*12.1*	*13.1*	*26.8*
		syn	**20.4**	28.0	10.8	34.3	50.1	35.4	13.3	23.8	18.4	14.6	3.5	23.0
	with DA	syn	18.6	36.1	15.1	35.1	54.0	**37.2**	**20.4**	**39.6**	26.5	**21.1**	15.1	29.0
		real	16.7	15.9	3.5	30.3	57.0	26.1	13.2	28.3	21.1	13.8	9.5	21.4
		joint	18.6	**37.1**	**15.4**	**38.0**	**57.7**	36.7	19.6	31.5	**36.5**	18.2	**16.5**	**29.6**

Table 5
Accuracy of the coarse-to-fine refinement with 24 fine viewpoints for the Pascal3D and ImageNet3D datasets. We compare the performance of our domain adaptation technique when taking all (*all*) or only non-occluded samples (*non-occ*).

		PASCAL3D (Xiang et al., 2014)											
		VGG CNN-pool5 - 24 views											
		Aero	Bike	Boat	Bus	Car	Chair	Table	mbike	Sofa	Train	TV	Avg.
	non-occ/all	*0.68*	*0.32*	*0.61*	*0.55*	*0.34*	*0.19*	*0.08*	*0.36*	*0.15*	*0.33*	*0.52*	*0.38*
all	w/o DA	24.3	30.7	18.2	43.3	36.1	26.2	22.2	32.7	25.6	30.0	29.6	29.0
	with DA	**29.0**	**39.4**	**26.5**	**46.1**	**40.4**	**30.8**	**27.7**	**38.9**	**38.2**	**37.8**	**37.4**	**35.7**
non-occ	w/o DA	29.2	33.3	15.4	**49.1**	**56.6**	32.5	20.0	31.1	41.1	**50.6**	35.1	35.8
	with DA	**32.2**	**43.9**	**24.5**	44.6	54.0	**45.0**	**21.0**	**53.5**	**50.6**	39.9	**40.1**	**40.8**
		ImageNet3D (Xiang et al., 2014)											
		VGG CNN-pool5 - 24 views											
		0.91	*0.56*	*0.93*	*0.95*	*0.94*	*0.94*	*0.31*	*0.50*	*0.44*	*0.81*	*0.95*	*0.75*
all	w/o DA	33.5	40.4	26.0	53.9	**63.5**	44.1	33.2	34.2	42.0	22.1	22.1	37.7
	with DA	**35.2**	**50.1**	**30.2**	**57.1**	63.2	**47.4**	**44.5**	**43.1**	**56.1**	**26.7**	**22.5**	**43.3**
non-occ	w/o DA	34.3	43.1	27.1	55.6	**64.4**	47.4	32.7	34.0	**44.9**	25.5	22.6	39.2
	with DA	**36.2**	**50.4**	**30.5**	**57.3**	64.1	**49.0**	**45.1**	**49.7**	41.9	**43.2**	**26.5**	**44.9**

improves the viewpoint accuracy by +8%. The results for ImageNet3D are reported in Table 8.

4.5. Cross-dataset viewpoint estimation

We finally perform a cross-dataset evaluation as in Sedaghat and Brox (2015). We evaluate the viewpoint estimation of cars from the Multi-View Car Dataset, Pascal3D, ImageNet3D and the dataset (Sedaghat and Brox, 2015), denoted as *Freiburg*, whose bounding boxes and viewpoints of cars were annotated in a fully unsupervised manner. The *Freiburg* dataset contains recorded scenes of 47 cars for a total of 5836 training images, on a full 360° rotation. For viewpoint estimation, we use the CNN approach (Tulsiani and Malik, 2015) as in Section 4.4 trained on the refined training data (*with*

Table 6
Pose estimation accuracy of our approach (*with DA joint*) for 24 fine viewpoints on the Pascal3D and ImageNet3D datasets. We compare the impact of training and testing with or without object occlusions.

Target		PASCAL3D (Xiang et al., 2014)											
		VGG CNN-pool5 - 24 views											
Train	Test	Aero	Bike	Boat	Bus	Car	Chair	Table	mbike	Sofa	Train	TV	Avg.
Non-occ	all	14.0	**19.8**	**9.8**	21.0	18.1	14.1	**16.0**	20.0	**24.8**	15.7	11.8	**16.8**
All		**18.4**	18.6	8.7	**24.1**	**20.7**	**17.0**	7.6	**21.2**	18.3	15.4	**14.2**	16.7
Non-occ	non-occ	16.8	18.1	**11.3**	**38.9**	33.1	**22.6**	**11.1**	**34.9**	18.3	**21.5**	**16.2**	**22.1**
All		**19.2**	**24.0**	8.7	24.1	**39.0**	17.0	7.6	21.0	18.3	15.4	14.2	19.0
		ImageNet3D (Xiang et al., 2014)											
		VGG CNN-pool5 - 24 views											
Non-occ	all	18.6	32.2	**15.4**	**38.1**	**57.3**	**37.0**	18.1	27.6	23.2	**18.9**	15.5	27.4
All		18.6	**37.1**	**15.4**	38.0	**57.3**	36.7	**19.6**	**31.5**	**36.5**	18.2	**16.5**	**29.6**
Non-occ	non-occ	18.6	35.3	**16.3**	36.9	59.1	38.3	**25.5**	35.0	30.4	**27.6**	15.6	30.8
All		19.0	**40.6**	15.8	**38.1**	**59.3**	**38.5**	24.0	**38.2**	36.7	22.0	**16.1**	**31.7**

Table 7
Pose estimation accuracies for the Pascal3D dataset using Tulsiani and Malik (2015) for regression and classification.

			PASCAL3D (Xiang et al., 2014)											
			MedError											
			aero	bike	boat	bus	car	chair	table	mbike	sofa	train	tv	Avg.
Regression	real	*gt*	*20.6*	*23.9*	*42.6*	*8.1*	*18.0*	*21.2*	*22.6*	*20.9*	*15.3*	*15.8*	*15.1*	*20.4*
		w/o DA	**26.5**	29.9	58.0	13.0	26.3	25.7	31.9	27.3	**19.7**	24.8	**16.1**	27.2
		with DA	32.2	**25.2**	**57.5**	**12.5**	**25.8**	**24.0**	**29.2**	**25.3**	21.6	**22.5**	17.9	**26.7**
	joint	w/o DA	**28.4**	25.0	**53.9**	**10.3**	23.8	25.8	**30.5**	22.5	18.6	23.2	15.5	25.2
		with DA	31.8	**20.6**	56.5	10.6	**22.3**	**24.9**	31.4	**20.1**	**15.9**	**19.0**	**13.8**	**24.3**
Classification	real	*gt*	*17.7*	*20.3*	*47.8*	*5.8*	*18.1*	*21.0*	*12.1*	*18.0*	*13.8*	*14.7*	*17.2*	*18.8*
		w/o DA	35.3	28.1	**52.3**	18.4	22.8	32.8	45.0	23.4	24.9	27.5	23.2	30.3
		with DA	**32.8**	**21.3**	70.4	**8.6**	**20.8**	**26.9**	**30.0**	**20.6**	**18.7**	**16.8**	**17.5**	**25.9**
	joint	w/o DA	**24.5**	20.0	**53.7**	7.2	18.1	25.6	**30.0**	21.3	**15.0**	15.0	20.5	22.8
		with DA	26.8	**17.5**	54.7	**6.9**	**16.5**	**23.3**	**30.0**	**18.3**	15.1	**14.6**	**16.3**	**21.8**
			24 views											
Regression	real	*gt*	*13.4*	*17.3*	*8.6*	*18.9*	*23.5*	*18.1*	*6.6*	*19.3*	*13.3*	*11.7*	*11.6*	*14.8*
		w/o DA	**14.3**	13.9	7.6	10.7	19.0	14.7	6.8	16.4	**19.3**	**9.4**	12.8	13.2
		with DA	13.2	**19.1**	**8.0**	**16.6**	**20.2**	**15.0**	**10.2**	**19.5**	18.9	**9.4**	**13.5**	**14.9**
	joint	w/o DA	13.2	19.1	**7.9**	16.6	20.2	15.0	**10.2**	19.5	**18.9**	9.4	12.5	14.8
		with DA	**13.6**	**19.8**	7.8	**21.6**	**22.1**	**17.3**	7.8	**24.8**	13.3	**10.8**	**18.4**	**16.1**
Classification	real	*gt*	*25.1*	*19.4*	*12.3*	*30.3*	*29.2*	*23.8*	*15.4*	*22.5*	*16.9*	*15.6*	*15.4*	*20.5*
		w/o DA	14.8	19.1	**8.8**	8.9	**26.2**	16.8	11.9	**22.0**	18.0	11.1	7.8	15.0
		with DA	**16.2**	**20.3**	5.0	**21.4**	24.9	**20.6**	**12.2**	19.4	**23.5**	**15.5**	**9.6**	**17.1**
	joint	w/o DA	**19.2**	25.2	10.8	33.7	**27.9**	23.7	16.2	21.4	21.6	18.4	12.8	21.0
		with DA	**19.2**	**27.2**	**14.9**	**35.7**	**27.9**	**24.8**	**18.9**	**27.4**	**33.8**	**19.8**	14.0	**24.0**
		DA LSVM	18.4	18.6	8.7	24.1	20.7	17.0	7.6	21.2	18.3	15.4	**14.2**	16.7

Table 8
Pose estimation accuracies for the ImageNet3D dataset using Tulsiani and Malik (2015) for classification.

			ImageNet3D (Xiang et al., 2014) CNN-classification											
			Aero	Bike	Boat	Bus	Car	Chair	Table	mbike	Sofa	Train	TV	Avg.
MedError	real	*gt*	*8.3*	*8.7*	*11.1*	*4.2*	*4.4*	*4.7*	*5.2*	*10.6*	*4.0*	*5.7*	*7.6*	*6.8*
		w/o DA	**20.2**	12.5	26.6	5.7	5.8	9.4	22.5	14.2	**6.9**	9.0	15.7	13.5
		with DA	22.5	**11.4**	**22.5**	**5.2**	**5.7**	**7.4**	**16.0**	**12.6**	7.5	**8.4**	**15.0**	**12.2**
	joint	w/o DA	**19.7**	10.1	22.7	5.6	5.6	8.3	20.6	12.5	**6.6**	8.9	14.4	12.3
		with DA	20.9	**8.7**	**21.6**	**5.1**	**5.5**	**6.9**	**15.2**	**11.5**	7.2	**8.5**	**13.5**	**11.3**
24 views	real	*gt*	*41.8*	*41.9*	*35.4*	*57.6*	*69.9*	*42.0*	*46.2*	*38.0*	*26.2*	*23.2*	*24.9*	*40.6*
		w/o DA	**24.8**	**36.6**	17.8	42.2	**60.7**	32.1	21.0	27.7	23.4	23.0	**24.9**	30.4
		with DA	22.4	36.3	**18.9**	**47.0**	59.7	**41.6**	**26.4**	**31.0**	**24.0**	**28.7**	23.3	**32.7**
	joint	w/o DA	**26.2**	42.6	**21.8**	46.2	62.2	29.7	21.3	37.7	30.5	27.8	25.1	33.7
		with DA	25.9	**46.8**	20.9	**51.0**	**62.3**	**47.9**	**26.8**	**39.3**	33.5	**28.0**	**30.5**	**37.5**
		DA LSVM	18.6	37.1	15.4	38.0	57.7	36.7	19.6	31.5	**36.5**	18.2	16.5	29.6

Table 9
Viewpoint estimation across datasets. The mean absolute error of viewpoint estimation (in degrees) is reported. In the cases denoted by *, Sedaghat and Brox (2015) uses the entire dataset for training while we use only the training data of the dataset.

		test			
train		Freiburg (Sedaghat and Brox, 2015)	Multi-View Car (Ozuysal et al., 2009)	Pascal3D (Xiang et al., 2014)	ImageNet3D (Xiang et al., 2014)
Freiburg (Sedaghat and Brox, 2015)	Sedaghat (Sedaghat and Brox, 2015)	–	34.4	61.5	38.0
	with DA		**21.5**	**58.0**	**27.3**
Multi-View Car (Ozuysal et al., 2009)	Sedaghat* (Sedaghat and Brox, 2015)	34.6	–	71.6	53.2
	with DA	**20.7**		**70.9**	**39.8**
Pascal3D (Xiang et al., 2014)	Sedaghat* (Sedaghat and Brox, 2015)	26.9	37.0	–	29.3
	with DA	**15.4**	**22.6**		**17.9**
ImageNet3D (Xiang et al., 2014)	Sedaghat (Sedaghat and Brox, 2015)	10.6	**17.4**	**47.7**	12.3
	with DA	**8.1**	18.7	51.0	**11.5**

DA) and compare it with the approach (Sedaghat and Brox, 2015). The results reported in Table 9 show that our approach performs very well across datasets. Our approach outperforms (Sedaghat and Brox, 2015) for 11 out of 13 configurations. For some dataset combinations, the mean absolute error is reduced by about 14 degrees compared to (Sedaghat and Brox, 2015).

5. Conclusions

In this work, we have presented an approach for weakly supervised domain adaptation for the task of viewpoint estimation. It uses synthetic data to refine the viewpoint annotations of the coarsely labelled training images. Using coarse viewpoint annotations of real images as weak supervision together with accurately annotated synthesized images is not only a very efficient approach to collect training data for fine-grained viewpoint estimation, it also allows to achieve an accuracy that goes beyond the abilities of human annotators. An extensive evaluation on five datasets for viewpoint estimation showed that our approach outperforms generic domain adaptation methods, proves effective for a large number of object classes and presents a considerable tolerance against occlusions.

Acknowledgements

The work was supported by the DFG project (GA 1927/2-2 FOR 1505).

References

Aytar, Y., Zisserman, A., 2011. Tabula rasa: model transfer for object category detection. IEEE International Conference on Computer Vision. pp. 2252–2259.

Baktashmotlagh, M., Harandi, M.T., Lovell, B.C., Salzmann, M., 2013. Unsupervised domain adaptation by domain invariant projection. IEEE International Conference on Computer Vision. pp. 769–776.

Busto, P., Liebelt, J., Gall, J., 2015. Adaptation of synthetic data for coarse-to-fine viewpoint refinement. British Machine Vision Conference.

Csurka, G., Chidlowskii, B., Clinchant, S., Michel, S., 2016. Unsupervised domain adaptation with regularized domain instance denoising. European Conference on Computer Vision. pp. 458–466.

Csurka, G., Dance, C., Fan, L., Willamowski, J., Bray, C., 2004. Visual categorization with bags of keypoints. Workshop on Statistical Learning in Computer Vision. pp. 1–22.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. IEEE Conference on Computer Vision and Pattern Recognition. pp. 886–893.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. ImageNet: a large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., Darrell, T., 2014. DeCAF: a deep convolutional activation feature for generic visual recognition. International Conference on Machine Learning. pp. 647–655.

Duan, L., Xu, D., Tsang, I., Luo, J., 2012. Visual event recognition in videos by learning from web data. IEEE Trans. Pattern Anal. Mach. Intell. 34 (9), 1667–1680.

Elhoseiny, M., El-Gaaly, T., Bakry, A., Elgammal, A., 2016. A comparative analysis and study of multiview CNN models for joint object categorization and pose estimation. International Conference on Machine Learning. pp. 888–897.

Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A., 2010. The Pascal visual object classes (VOC) challenge. Int. J. Comput. Vis. 88 (2), 303–338.

Fanelli, G., Dantone, M., Gall, J., Fossati, A., Van Gool, L., 2013. Random forests for real time 3D face analysis. Int. J. Comput. Vis. 101 (3), 437–458.

Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D., 2010. Object detection with discriminatively trained part-based models. IEEE Trans. Pattern Anal. Mach. Intell. 32 (9), 1627–1645.

Fenzi, M., Leal-Taixe, L., Rosenhahn, B., Ostermann, J., 2013. Class generative models based on feature regression for pose estimation of object categories. IEEE Conference on Computer Vision and Pattern Recognition. pp. 755–762.

Fernando, B., Habrard, A., Sebban, M., Tuytelaars, T., 2013. Unsupervised visual domain adaptation using subspace alignment. IEEE International Conference on Computer Vision. pp. 2960–2967.

Fidler, S., Dickinson, S., Urtasun, R., 2012. 3D object detection and viewpoint estimation with a deformable 3D cuboid model. Advances in Neural Information Processing Systems. pp. 611–619.

Ganin, Y., Lempitsky, V., 2015. Unsupervised domain adaptation by backpropagation. International Conference on Machine Learning. pp. 1180–1189.

Geiger, A., Lenz, P., Urtasun, R., 2012. Are we ready for autonomous driving? The KITTI vision benchmark suite. IEEE Conference on Computer Vision and Pattern Recognition. pp. 3354–3361.

Ghifary, M., Kleijn, W.B., Zhang, M., 2014. Domain adaptive neural networks for object recognition. Pacific Rim International Conference on Artificial Intelligence. pp. 898–904.

Ghodrati, A., Pedersoli, M., Tuytelaars, T., 2014. Is 2D information enough for viewpoint estimation? British Machine Vision Conference. pp. 1–12.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587.

Glasner, D., Galun, M., Alpert, S., Basri, R., Shakhnarovich, G., 2011. Aware object detection and pose estimation. IEEE International Conference on Computer Vision. pp. 1275–1282.

Glasner, D., Galun, M., Alpert, S., Basri, R., Shakhnarovich, G., 2012. Aware object detection and continuous pose estimation. Image Vis. Comput. 30 (12), 923–933.

Gong, B., Grauman, K., Sha, F., 2013. Reshaping visual datasets for domain adaptation. Advances in Neural Information Processing Systems. pp. 1286–1294.

Gong, B., Shi, Y., Sha, F., Grauman, K., 2012. Geodesic flow kernel for unsupervised domain adaptation. IEEE Conference on Computer Vision and Pattern Recognition. pp. 2066–2073.

Gopalan, R., Li, R., Chellappa, R., 2011. Domain adaptation for object recognition: an unsupervised approach. IEEE Conference on Computer Vision and Pattern Recognition. pp. 999–1006.

Gu, C., Ren, X., 2010. Discriminative mixture-of-templates for viewpoint classification. European Conference on Computer Vision. pp. 408–421.

He, K., Sigal, L., Sclaroff, S., 2014. Parameterizing object detectors in the continuous pose space. European Conference on Computer Vision. pp. 450–465.

Hejrati, M., Ramanan, D., 2014. Analysis by synthesis: 3D object recognition by object reconstruction. IEEE Conference on Computer Vision and Pattern Recognition. pp. 2449–2456.

Hoffman, J., Rodner, E., Donahue, J., Saenko, K., Darrell, T., 2013. Efficient learning of domain-invariant image representations. International Conference on Learning Representations.

Jhuo, I., Liu, D., Lee, D., Chang, S., 2012. Robust visual domain adaptation with low-rank reconstruction. IEEE Conference on Computer Vision and Pattern Recognition. pp. 2168–2175.

Johnson, S. G., The NLopt nonlinear-optimization package, https://nlopt.readthedocs.io/en/latest/Citing_NLopt/.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. ImageNet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems.

pp. 1097–1105.

Kuhn, H.W., 1955. The Hungarian method for the assignment problem. Nav. Res. Log. Q. 2 (1–2), 83–97.

Leibe, B., Leonardis, A., Schiele, B., 2004. Combined object categorization and segmentation with an implicit shape model. European Conference on Computer Vision. pp. 17–32.

Liebelt, J., Schmid, C., 2010. Multi-view object class detection with a 3D geometric model. IEEE Conference on Computer Vision and Pattern Recognition. pp. 1688–1695.

Long, M., Zhu, H., Wang, J., Jordan, M.I., 2016. Unsupervised domain adaptation with residual transfer networks. Advances in Neural Information Processing Systems. pp. 136–144.

Marín, J., Vázquez, D., Gerónimo, D., López, A., 2010. Learning appearance in virtual scenarios for pedestrian detection. IEEE Conference on Computer Vision and Pattern Recognition. pp. 137–144.

Massa, F., Aubry, M., Marlet, R., 2014. Convolutional neural networks for joint object detection and pose estimation: a comparative study. CoRR. arXiv:1412.7190.

Massa, F., Marlet, R., Aubry, M., 2016. Crafting a multi-task CNN for viewpoint estimation. British Machine Vision Conference.

Matzen, K., Snavely, N., 2013. NYC3DCars: a dataset of 3D vehicles in geographic context. IEEE International Conference on Computer Vision. pp. 761–768.

Mottaghi, R., Xiang, Y., Savarese, S., 2015. A coarse-to-fine model for 3D pose estimation and sub-category recognition. IEEE Conference on Computer Vision and Pattern Recognition. pp. 418–426.

Ozuysal, M., Lepetit, V., Fua, P., 2009. Pose estimation for category specific multiview object localization. IEEE Conference on Computer Vision and Pattern Recognition. pp. 778–785.

Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q., 2011. Domain adaptation via transfer component analysis. IEEE Trans. Neural Netw. 22 (2), 199–210.

Peng, X., Sun, B., Ali, K., Saenko, K., 2015. Learning deep object detectors from 3d models. IEEE International Conference on Computer Vision. pp. 1278–1286.

Pepik, B., Stark, M., Gehler, P., Ritschel, T., Schiele, B., 2015. 3D object class detection in the wild. IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1–10.

Pepik, B., Stark, M., Gehler, P., Schiele, B., 2012. Teaching 3D geometry to deformable part models. IEEE Conference on Computer Vision and Pattern Recognition. pp. 3362–3369.

Phong, B.T., 1975. Illumination for computer generated pictures. Commun. ACM 18 (6), 311–317.

Pishchulin, L., Jain, A., Wojek, C., Andriluka, M., Thormählen, T., Schiele, B., 2011. Learning people detection models from few training samples. IEEE Conference on Computer Vision and Pattern Recognition. pp. 1473–1480.

Redondo-Cabrera, C., López-Sastre, R., Tuytelaars, T., 2014. All together now: simultaneous object detection and continuous pose estimation using a Hough forest with probabilistic locally enhanced voting. British Machine Vision Conference.

Saenko, K., Kulis, B., Fritz, M., Darrell, T., 2010. Adapting visual category models to new domains. European Conference on Computer Vision. pp. 213–226.

Savarese, S., Fei-Fei, L., 2007. 3D generic object categorization, localization and pose estimation. IEEE International Conference on Computer Vision. pp. 1–8.

Schels, J., Liebelt, J., Lienhart, R., 2012. Learning an object class representation on a continuous viewsphere. IEEE Conference on Computer Vision and Pattern Recognition. pp. 3170–3177.

Sedaghat, N., Brox, T., 2015. Unsupervised generation of a viewpoint annotated car dataset from videos. IEEE International Conference on Computer Vision. pp. 1314–1322.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. CoRR. arXiv:1409.1556.

Stark, M., Goesele, M., Schiele, B., 2010. Back to the future: learning shape models from 3D CAD data. British Machine Vision Conference.

Su, H., Qi, C.R., Li, Y., Guibas, L.J., 2015. Render for CNN: viewpoint estimation in images using CNNs trained with rendered 3D model views. IEEE International Conference on Computer Vision. pp. 2686–2694.

Sun, B., Feng, J., Saenko, K., 2015. Return of frustratingly easy domain adaptation. AAAI Conference on Artificial Intelligence. pp. 2058–2065.

Sun, B., Saenko, K., 2014. From virtual to reality: fast adaptation of virtual object detectors to real domains. British Machine Vision Conference.

Svanberg, K., 2002. A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J. Optim. 12 (2), 555–573.

Torki, M., Elgammal, A., 2011. Regression from local features for viewpoint and pose estimation. IEEE International Conference on Computer Vision. pp. 2603–2610.

Tulsiani, S., Malik, J., 2015. Viewpoints and keypoints. IEEE Conference on Computer Vision and Pattern Recognition. pp. 1510–1519.

Tzeng, E., Hoffman, J., Darrell, T., Saenko, K., 2015. Simultaneous deep transfer across domains and tasks. IEEE International Conference on Computer Vision. pp. 4068–4076.

Vázquez, D., López, A., Marín, J., Ponsa, D., Gerónimo, D., 2014. Virtual and real world adaptation for pedestrian detection. IEEE Trans. Pattern Anal. Mach. Intell. 36 (4), 797–809.

Vázquez, D., López, A., Ponsa, D., Marín, J., 2011. Cool world: domain adaptation of virtual and real worlds for human detection using active learning. Advances in Neural Information Processing Systems: Workshop on Domain Adaptation: Theory and Applications.

Xiang, Y., Mottaghi, R., Savarese, S., 2014. Beyond Pascal: a benchmark for 3D object detection in the wild. IEEE Winter Conference on Applications of Computer Vision. pp. 75–82.

Xu, Z., Li, W., Niu, L., Xu, D., 2014. Exploiting low-rank structure from latent domains for domain generalization. European Conference on Computer Vision. pp. 628–643.

Yang, J., Yan, R., Hauptmann, A.G., 2007. Cross-domain video concept detection using adaptive SVMs. ACM International Conference on Multimedia. pp. 188–197.

Zia, M.Z., Stark, M., Schiele, B., Schindler, K., 2013. Detailed 3D representations for object recognition and modeling. IEEE Trans. Pattern Anal. Mach. Intell. 35 (11), 2608–2623.