

Adaptation of Synthetic Data for Coarse-to-Fine Viewpoint Refinement

Pau Panareda Busto^{1,2}
 pau.panareda-busto@airbus.com
 Joerg Liebelt¹
 joerg.liebelt@airbus.com
 Juergen Gall²
 gall@iai.uni-bonn.de

¹ Airbus Group Innovations
 Munich, Germany
² Computer Vision Group
 Institute of Computer Science III
 University of Bonn, Germany

Abstract

The quality of learning-based pose estimation still heavily relies on manual training data annotations. However, the manual labeling of large datasets is costly and frequently limited to a few coarse viewpoint annotations of varying accuracy. In this work, we propose to refine such coarse pose annotations with a domain adaptation approach, where the source domain consists of fine-grained pose annotations generated from synthetic computer graphics models, and the target domain of coarse manual pose annotations of a real dataset. Our domain adaptation step computes a linear map which aligns corresponding samples from the two domains and allows for the refinement of the manual pose labels using the transformed synthetic ones. Experiments show that we significantly improve pose estimation on several state-of-the-art car datasets.

1 Introduction

Although pose estimation of object classes is an important task in scene understanding, most methods consider only the estimation of coarse viewpoints, namely front, back, left and right view. The main reason is the lack of training data of images with continuous or fine-grained viewpoint annotations. As illustrated in Figure 1, humans perform poorly for estimating the viewpoint of an object accurately, but they perfectly estimate the four coarse viewpoints. An alternative is the use of synthetic data, which has been successfully used to augment real training images for object detectors [16, 22, 27, 29, 30]. In this case, however, the real data can be annotated by humans and the synthetic data only increases the variation in the training data, but does not refine the labels of the training data. If only synthetic data is used, the viewpoint annotations are even continuous [18], but the training data lacks the realism of real images, which results in a loss in accuracy. The limitation of synthetic data is the cost of acquiring detailed 3D models of object classes and render them in various environments. Examples of our synthetic images are shown in Figure 2(a).

In this work, we propose to leverage human annotators and synthetic data to avoid the fine annotation of images by annotators, 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,

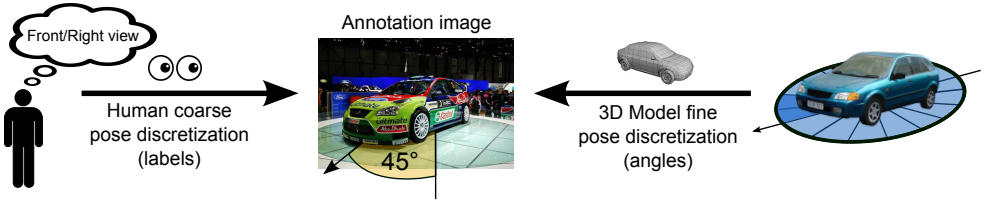


Figure 1: Humans are perfect for annotating coarse viewpoints of objects in real images, but fail to estimate pose accurately at a fine level. 3D graphic models can be used to synthesize data at very accurate fine angles, but it is time-consuming to model all appearance variations present in real images. We therefore propose to leverage the abilities of humans of estimating coarse viewpoints and the pose accuracy of synthetic data.

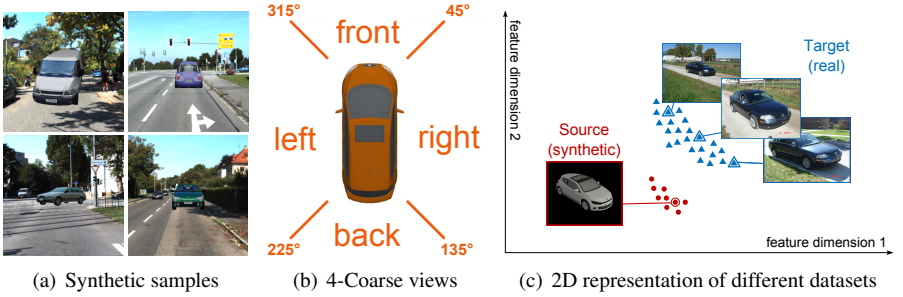


Figure 2: (a) Some examples of our synthetic images, whose rendered cars implicitly contain fine annotations. (b) The four views available for real images. (c) Synthetic and real images with the same annotated viewpoint lie in different domains within the feature space.

sketched in Figure 2(b), and we propose an approach that refines the labels using synthetic data. Since synthetic data and real images are different domains as illustrated in Figure 2(c), we use a domain adaptation approach for the refinement. Our experiments show that standard domain adaptation approaches like [0, 0] are not sufficient for label refinement. Instead, we propose an approach that exploits the coarse labels of the real training images.

In the experiments on four car datasets for viewpoint estimation, we show that our approach with domain adaptation performs better than using only synthetic data and it outperforms other domain adaptation methods. For some datasets, the achieved accuracy is even comparable to the accuracy of a viewpoint estimator trained on real images with fine viewpoint annotations.

2 Related Work

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 [0, 0, 0] or the so-called source domain is used to regularize the learning of a classifier on the target domain [0, 0]. A popular choice in this context are support vector machines [0, 0, 0, 0, 0]. The approaches that estimate the transformations without a classifier like the geodesic flow kernel [0] 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 unlabeled, or in a supervised or semi-supervised setting where the data from the target domain contains a few labeled samples. In contrast to these works, we use domain adaptation in a weakly supervised setting where only coarse labels are available for at least some of the images of the target domain.

During the last years, the main focus has been on the optimization process for domain adaptation, where additional constraints for the optimization have been proposed [10, 9, 11, 12, 13]. For instance, orthogonality constraints have been suggested for the transformation matrix [10, 11]. Other approaches paid more attention to relaxation techniques to make the optimization solvable [10, 13]. While these methods assume that the source and target domains are known, [8] calculates latent domains from the given annotated datasets.

Synthetic Data The use of 3D models to estimate the viewpoint of object instances has been addressed in several works [14, 15, 16, 17, 18, 19]. In these works, 3D models are used to learn the spatial 3D relations of parts or features. In contrast to these works, we use 3D models to synthesize training images with accurate viewpoint annotations.

Synthetic data has also been used in the context of pedestrian detection [16, 17]. While [16] uses only synthetic data for training, [17] uses it in addition to real images of pedestrians. Since the synthetic and the real images have the same labels, both training sets are combined by training a classifier for each dataset and combining them by another classifier on top of them. Recently, datasets consisting of images annotated with 3D models have been proposed [11, 13]. Considering that the manual annotation is very time consuming, we do not assume that the real images are accurately annotated. Instead, we use the synthetic data to refine the coarsely labeled real images. The discrepancy between real and synthetic images was addressed in [17, 19, 20]. In [19, 20], an active learning approach is proposed. To this end, a pedestrian classifier is trained on the synthetic data and applied to the real training images. The misclassified examples in the real images are then manually selected and used as additional training images. In [20] whitening is applied to the synthetic images.

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 [11].

3 Domain Adaptation for Viewpoint Refinement

Since synthetic data and real images belong to different domains as illustrated in Figure 2(c), 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 pose classifiers, *i.e.*, linear support vector machines (SVM), trained on the transformed synthetic data.

The learning of the mapping from the source to the target domain is discussed in Section 3.1. The establishment of correspondences between clusters of both domains is discussed in Section 3.2. Finally, Section 3.3 discusses the label refinement.

3.1 Domain Adaptation

To map the source data to the target domain, we have to learn a mapping from $\mathcal{S} \in \mathbb{R}^D$ to $\mathcal{T} \in \mathbb{R}^D$, where D denotes the dimensionality of the features. For label refinement, the

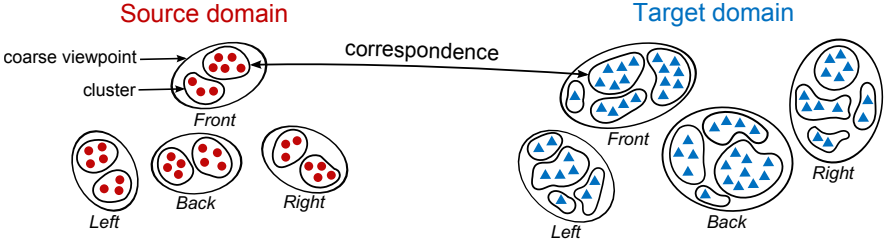


Figure 3: Each cluster in the target domain is assigned to a source cluster that belongs to the same coarse viewpoint. In this example, for an 8-view refinement: $V_i = 2$ and $K_i = 4$.

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 $S = \{s_1, \dots, s_M\}$ and $T = \{t_1, \dots, t_N\}$ 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 $C = \{c_1, \dots, c_K\}$ with (s_{c_k}, t_k) and $K \leq N$ will be explained in Section 3.2.

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

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

which can be expressed in matrix form:

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

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. The objective is equivalent to the spectral norm of $WP_S - P_T$ and can be solved by taking the largest singular value.

The objective, however, can be faster optimized by non-linear optimization. To this end, the derivatives of (2) are calculated by

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

In our implementation, we use the local gradient-based optimization method of moving asymptotes [28], which is part of the NLOPT package [29].

3.2 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 $i = \{\text{front, back, left, right}\}$, *i.e.*, $V = \sum_i V_i$, where V is the total number of fine-grained poses. For the target domain, we only have the coarse viewpoints and therefore cluster the N_i training samples of one 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 $V_i \leq K_i \leq N_i$. For the clustering, we represent each image by a HOG feature vector and append the aspect ratio of the bounding box surrounding the object.

As illustrated in Figure 3, 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 $\hat{S}^i = \{\hat{s}_1^i, \dots, \hat{s}_{V_i}^i\}$ and $\hat{T}^i = \{\hat{t}_1^i, \dots, \hat{t}_{K_i}^i\}$. The correspondences are then established by solving a bipartite matching problem:

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

It assigns to each cluster in the target domain a unique cluster in the source domain. Since there can be more clusters in the target domain than in the source domain, each source is associated to $a_v = \frac{K_i}{V_i}$ 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 [14] to solve the problem and for any cluster pair with $e_{vk} = 1$, we obtain a correspondence c . The correspondences from all coarse views are then used to estimate the transformation W in (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 $v = \{1, \dots, V\}$. 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:

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

where w_v and b_v are the weights and bias of the linear SVM for the fine viewpoint v .

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 an 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 4 well-known car datasets with annotated poses. The *3D Obj. Categorization* [24] dataset provides 10 image sets of cars in 8 different angles (every 45 degrees), permitting a refinement from 4 to 8 fine viewpoints. There are 2 elevations and 3 distances for each view, giving 48 images per car. We take 7 sets for training and 3 for testing. The *EPFL* [19] dataset contains sequences of 20 cars as they rotate by 360 degrees, where one image is taken every 3-4 degrees. These fine-grained poses allow us to test the refinement for 8, 16 and 32 fine viewpoints. We take the first 10 car sequences as training (1179 images) and the last 10 as test data (1120 images). All cars in these two datasets are in

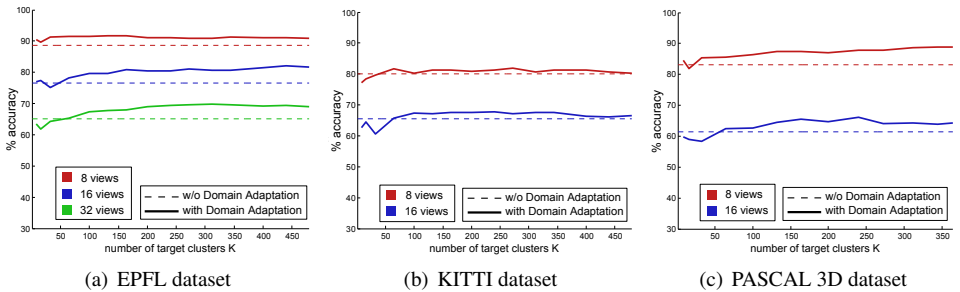


Figure 4: Impact of the number of target clusters K for viewpoint refinement.

a fixed location with distinctive details. Therefore, we also evaluate our method on the more challenging *KITTI* [6] 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 17463 cars, 7811 of which are non-occluded. The last is the *PASCAL3D* [6] dataset that enriches the PASCAL VOC 2012 [6] categories with 3D annotations and provides a car training set (529 images, 364 non-occluded) and a validation set (477 images, 319 non-occluded).

The setup for the experiments is as follows. We automatically generate synthetic data of 15 textured car models with random background images taken from the *KITTI* dataset [6]. A background image is discarded when the rendered car model overlaps with a car in the image. These images, which point towards the car’s driving direction, allow for synthetic car placements in the center of the image with an acceptable level of coherence without any human supervision throughout the process. The synthetic images are then obtained by rotating the car models every 5 degrees (72 azimuth angles), using 3 levels of elevation (0, 15 and 25 degrees) and 2 different distances. This results in 6480 fully annotated synthetic samples as shown in Figure 2(a). The pose labels are quantized to their closest angle of the V fine poses. For the real training images, we use the bounding boxes and convert the given viewpoints into the four coarse views for the refinement task, that is: *front* = $(315^\circ, \dots, 45^\circ)$, *right* = $[45^\circ, \dots, 135^\circ]$, *back* = $(135^\circ, \dots, 225^\circ)$ and *left* = $[225^\circ, \dots, 315^\circ]$. For the real test images we use the given bounding boxes if the images are not already cropped. Neither coarse nor fine viewpoints are used for the test images. As training and testing, we rescale the bounding boxes to 128×128 pixels and extract HOG descriptors [9] with 8 bins, forming feature vectors of 2976 dimensions. We additionally normalize the synthetic and the real feature vectors separately such that the mean is zero and the standard deviation one.

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.

Impact of number of target clusters. As described in Section 3.2, 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 different numbers of the

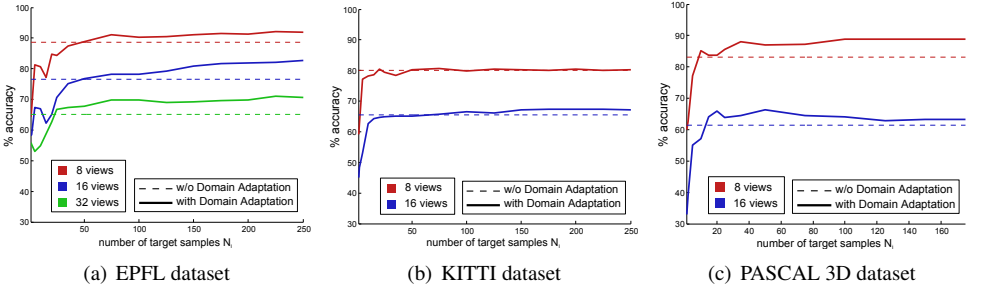


Figure 5: Impact of the number of target samples N_i per coarse view for the refinement.

fine viewpoints used for evaluation are shown in Figure 4. 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.

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 labeled target samples N . To avoid any clustering artifacts, 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 Figure 5 show that already 50-75 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.

Accuracy of the viewpoint refinement We finally compare the refinement accuracy of our method with different popular domain adaptation techniques [9, 10, 27]. These techniques follow the same strategy of transforming domains. Thus, we also train and apply our linear SVMs on the transformed domains for label refinement as described in Section 3.3. As baseline, we use the linear SVMs trained on the synthetic data without domain adaptation. In [27], a whitening transformation is applied to both domains. The geodesic flow kernel (GFK) [9] is an unsupervised domain adaptation method that maps both domains to a common subspace in a Grassmannian manifold. The approach can also be used for supervised domain adaptation, but it did not improve the results in our experiments. We therefore report the results for the unsupervised approach. The maximum margin domain transform (MMDT) [10] is a supervised domain adaptation approach that computes the linear SVMs and the domain transformation in an iterative process where in each iteration either the SVMs or the transformation are estimated. We achieved the best performance for this approach when using the coarse viewpoint labels for the synthetic and the real images.

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 pose estimation as in the previous experiments. We report numbers for $K = V$, $K = 100$ and $K = N$, *i.e.*, each real training sample is one cluster. For the first two settings we report the mean accuracy and its standard deviation over 10 runs since K-Means clustering contains slight variations depending on its initialization. The results for the different datasets are shown in Table 1. While $K = N$ performs best for three out of four datasets, $K = 100$ performs best for the KITTI dataset. The KITTI dataset contains the largest number of real training images and









	3DObj 	EPFL 				KITTI 		PASCAL3D 	
	8views	8views	16views	32views	8views	16views	8views	16views	
w/o Dom. Adapt.	97.62	88.65	76.63	65.14	80.04	65.61	83.09	61.36	
Whitening	99.11	87.98	79.64	67.08	78.03	62.23	84.34	66.36	
GFK 	97.62	88.92	76.58	65.31	80.07	65.42	84.97	59.67	
MMDT 	97.92	88.25	77.02	64.68	78.94	62.37	84.80	60.21	
V=views, K=V	98.57 (0.53)	90.41 (1.65)	77.32 (2.01)	64.19 (1.86)	77.15 (1.30)	64.56 (1.67)	84.55 (1.94)	58.89 (3.01)	
V=views, K=100	99.11 (0.36)	91.57 (0.47)	79.62 (0.65)	67.40 (1.18)	80.32 (1.41)	67.37 (1.47)	86.53 (1.83)	62.67 (1.82)	
V=views, K=N	99.70	92.00	81.82	70.41	78.78	67.05	88.79	63.33	
V=M, K=N	92.86	85.69	76.69	65.49	75.70	62.92	80.46	58.77	

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.

reducing the samples by clustering is beneficial in this case. In addition, this dataset has the specific feature that it is dominated by 4 out of 16 viewpoints: front and back views for vehicles in the same road and side views for the crossing ones. Therefore, fine viewpoints from the synthetic dataset yield only minor improvements. 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. This shows that the synthetic data needs to be quantized according to the fine-grained views.

Table 1 also reports the numbers for the other domain adaptation methods. In both settings, $K = 100$ or $K = N$, our approach outperforms the other methods for pose refinement. Only for the 16 views on the PASCAL3D dataset, whitening performs better. Our approach with $K = 100$ also performs better than the baseline without domain adaptation for all datasets whereas the approaches  and  are in few cases even slightly below the baseline.

4.2 Viewpoint Estimation

We finally 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, the real training data with refined viewpoint labels or on both datasets. For the refinement, we use our approach with $K = N$ (*with DA*) and compare it to the refinement without domain adaptation (*w/o DA*), *i.e.*, using only the linear SVMs as described in Section 3.3. We also evaluate the accuracy when the real images are refined directly by the established correspondences (*real corr*), as described in Section 3.2, but without estimating the transformation W . We report the results in Table 2 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 expected upper bound of the accuracy in comparison to the other settings, where only the four coarse viewpoint labels of the real training images are used.

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 pose estimation for all datasets. When we only use the correspondences for refinement of the real training images without transforming the synthetic data (*real corr*), the results are in few cases better than our approach with domain adaptation (*with DA real*), but for the KITTI dataset the accuracy is much worse. This is due to the aforementioned coarse-view dominance, whose lack of fine viewpoints in the target domain lead to erroneous





		3DObj 	EPFL 			KITTI 		PASCAL3D 	
		8views	8views	16views	32views	8views	16views	8views	16views
w/o DA	<i>gt</i>	99.31	80.06	73.57	60.59	82.23	77.89	56.84	35.41
	syn	75.69	65.98	60.92	46.55	58.69	47.25	56.50	49.53
	real	99.31	76.04	65.46	49.90	74.43	55.69	53.21	35.43
	joint	88.89	72.52	63.81	50.04	72.75	54.30	58.31	53.78
	real corr	99.31	77.43	70.25	58.53	63.44	51.71	52.54	40.49
with DA	syn	90.97	74.62	67.01	51.06	64.28	54.07	61.60	55.65
	real	99.31	78.37	69.04	55.22	74.46	56.28	54.56	39.17
	joint	93.06	75.73	71.93	53.00	73.23	59.04	61.60	55.28

Table 2: Pose estimation accuracy on test data using real training data, synthetic data or both training sets.

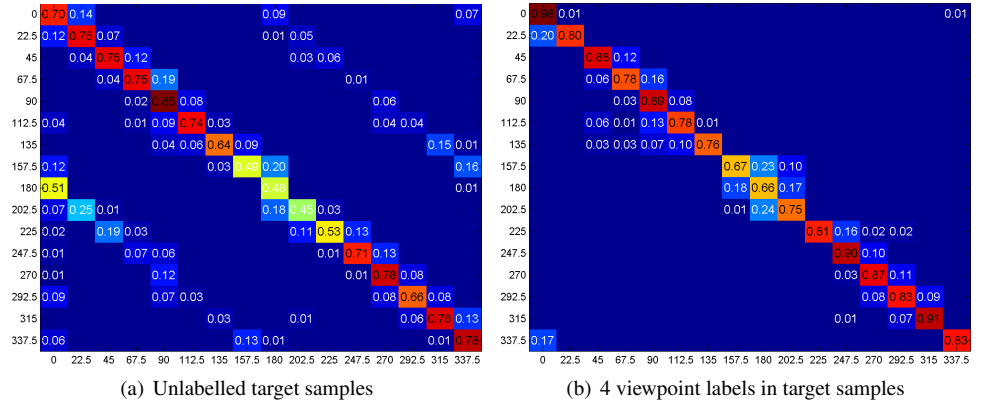


Figure 6: Confusion matrix for EPFL dataset in a 16-viewpoint refinement. (a) Without supervision rotations by 180 degrees are sometimes confused. (b) When weak supervision from the four coarse viewpoint labels is used, these confusions are resolved.

correspondences, while the global transformation matrix W attenuates their impact. Using the synthetic data not only for label refinement but also as additional training data for pose estimation (*with DA joint*) does not always improve the pose estimation accuracy. An exception is the PASCAL3D dataset. Since the training data does not contain many object class variations, the synthetic data significantly boosts the performance since it increases the shape variations within the training data.

Unsupervised refinement. Our approach could also be applied in an unsupervised setting where the real training images are not annotated. In this case, the clustering and the computation of the correspondences are no longer constrained by the coarse viewpoint labels as illustrated in Figure 3. The difference between the unsupervised and the weakly supervised setting with four coarse viewpoints is illustrated in Figure 6. Without supervision the confusion matrix shows three diagonals since two views that differ by 180 degrees are sometimes confused. Using weak supervision resolves this problem. This shows how our approach leverages the abilities of humans of estimating coarse viewpoints and the pose accuracy of synthetic data as illustrated in Figure 1.

5 Conclusions

In this paper, we have proposed to use synthetic data to refine the labels of real training images. We have evaluated our approach in the context of pose estimation, where the real images are manually labeled by only four coarse views, but finer viewpoint estimates are required. Due to the differences between the real and the synthetic data, we apply domain adaptation to align both domains and improve the viewpoint refinement. For domain adaptation, we consider the real images as weakly labeled data and use the coarse views to constrain the learning of the transformation from the synthetic data to the real data. We have evaluated our approach on four car datasets for pose estimation and compared it to other domain adaptation approaches. The results have shown that 3D generated models can be successfully used to refine labels in real images and therefore overcome the cumbersome annotation of real images by accurate and fine viewpoints. In particular, our approach leverages the abilities of humans of estimating coarse viewpoints and the pose accuracy of synthetic data.

Acknowledgements

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

References

- [1] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *IEEE International Conference on Computer Vision*, pages 2252–2259, 2011.
- [2] M. Baktashmotlagh, M. T. Harandi, B. C. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *IEEE International Conference on Computer Vision*, pages 769–776, 2013.
- [3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.
- [4] L. Duan, D. Xu, I. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1667–1680, 2012.
- [5] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The Pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.
- [6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.
- [7] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2066–2073, 2012.
- [8] B. Gong, K. Grauman, and F. Sha. Reshaping visual datasets for domain adaptation. In *Advances in Neural Information Processing Systems*, pages 1286–1294, 2013.

- [9] R. Gopalan, R. Li, and R. Chellappa. Domain adaptation for object recognition: An unsupervised approach. In *IEEE International Conference on Computer Vision*, pages 999–1006, 2011.
- [10] M. Hejrati and D. Ramanan. Analysis by synthesis: 3D object recognition by object reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2449–2456. IEEE, 2014.
- [11] J. Hoffman, E. Rodner, J. Donahue, B. Kulis, and K. Saenko. Asymmetric and category invariant feature transformations for domain adaptation. *International Journal of Computer Vision*, 109(1-2):28–41, 2014.
- [12] I. Jhuo, D. Liu, D. Lee, and S. Chang. Robust visual domain adaptation with low-rank reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2168–2175, 2012.
- [13] S. G. Johnson. The nlopt nonlinear-optimization package. <http://ab-initio.mit.edu/nlopt>, 2007–2010.
- [14] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
- [15] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1688–1695, 2010.
- [16] J. Marín, D. Vázquez, D. Gerónimo, and A. López. Learning appearance in virtual scenarios for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 137–144, 2010.
- [17] K. Matzen and N. Snavely. Nyc3dcars: A dataset of 3D vehicles in geographic context. In *IEEE International Conference on Computer Vision*, pages 761–768, 2013.
- [18] R. Mottaghi, Y. Xiang, and S. Savarese. A coarse-to-fine model for 3D pose estimation and sub-category recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 418–426, 2015.
- [19] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 778–785, 2009.
- [20] S. J. Pan, I. W. Tsang, J. T. Kwok, Q. Yang, et al. Domain adaptation via transfer component analysis. *IEEE Transactions on Neural Networks*, 22(2):199–210, 2011.
- [21] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3D geometry to deformable part models. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3362–3369, 2012.
- [22] L. Pishchulin, A. Jain, C. Wojek, M. Andriluka, T. Thormählen, and B. Schiele. Learning people detection models from few training samples. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1480, 2011.

- [23] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *IEEE European Conference on Computer Vision*, pages 213–226, 2010.
- [24] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007.
- [25] J. Schels, J. Liebelt, and R. Lienhart. Learning an object class representation on a continuous viewsphere. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3170–3177, 2012.
- [26] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3D cad data. In *British Machine Vision Conference*, volume 2, page 5, 2010.
- [27] B. Sun and K. Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *British Machine Vision Conference*, 2014.
- [28] K. Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555–573, 2002.
- [29] D. Vázquez, A. López, D. Ponsa, and J. Marín. Cool world: domain adaptation of virtual and real worlds for human detection using active learning. In *Advances in Neural Information Processing Systems, Workshop on Domain Adaptation: Theory and Applications*, 2011.
- [30] D. Vázquez, A. López, J. Marín, D. Ponsa, and D. Gerónimo. Virtual and real world adaptation for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):797–809, 2014.
- [31] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond Pascal: A benchmark for 3D object detection in the wild. In *IEEE Winter Conference on Applications of Computer Vision*, pages 75–82, 2014.
- [32] Z. Xu, W. Li, L. Niu, and D. Xu. Exploiting low-rank structure from latent domains for domain generalization. In *IEEE European Conference on Computer Vision*, pages 628–643, 2014.
- [33] J. Yang, R. Yan, and A. G. Hauptmann. Cross-domain video concept detection using adaptive svms. In *ACM International Conference on Multimedia*, pages 188–197, 2007.
- [34] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3D representations for object recognition and modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2608–2623, 2013.