OPEN ACCESS
OPUS AUGSBURG

Universität Augsburg
Universitätsbibliothek

Learning an object class representation on a continuous viewsphere

J. Schels, J. Liebelt, Rainer Lienhart

Angaben zur Veröffentlichung / Publication details:

Schels, J., J. Liebelt, and Rainer Lienhart. 2012. "Learning an object class representation on a continuous viewsphere." In *IEEE Conference on Computer Vision and Pattern Recognition, 16 - 21 June 2012, Providence, RI, USA*, 3170–77. Piscataway, NJ: IEEE. https://doi.org/10.1109/cvpr.2012.6248051.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

Learning an Object Class Representation on a Continuous Viewsphere

Johannes Schels*, Joerg Liebelt*
EADS Innovation Works
München, Germany
{johannes.schels,joerg.liebelt}@eads.net

Rainer Lienhart
University of Augsburg
Augsburg, Germany
lienhart@informatik.uni-augsburg.de

Abstract

We propose an approach to multi-view object class detection and approximate 3D pose estimation. It relies on CAD models as positive training examples and discriminatively learns photometric object parts such that an optimal coverage of intra-class and viewpoint variation is guaranteed. In contrast to previous work, the approach shows a significantly reduced training set dependency while avoiding any manual training supervision or annotation, since it is capable of deriving all relevant information exclusively from the provided set of 3D CAD models and an arbitrary set of 2D negative images. In entirely circumventing semantic or view-based representations, part symmetries and co-occurrences between viewpoints can be efficiently exploited. This, in turn, leads to a significantly lower complexity while still achieving state-of-the-art performance on two current benchmark data sets for two different object classes.

1. Introduction

In recent years, multi-view object class detection and approximate 3D pose estimation from single images have regained attention [10, 13, 21, 22, 23, 27]. Although the increasingly sophisticated representations suggested in previous work have improved detection and estimation precision, they typically have not led to a significantly better generalization power, nor reduced the degree of training supervision. To the contrary, the dependency on training set characteristics seems to have increased, e.g. in relying on structure from motion to build precise representations of the objects in the training set [10] or in assembling a set of view-dependent edge templates for each object instance [18]. Similarly, the level of training supervision has increased, requiring at least viewpoint labels [13, 23], sometimes even manual annotations of 3D correspondences [27]. In most cases, preference is also given to view-based representations [13, 20, 21, 27] or semantically chosen parts [21], which are trained separately, even though they frequently share significant geometrical and visual similarities. Only [14] recently considered the trade-off between class and viewpoint variation in a more principled way, although they require video sequences for training and testing.

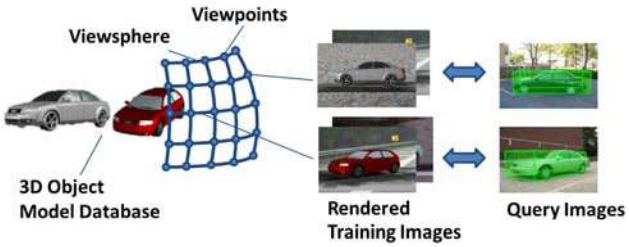

Figure 1. An object class representation covering the entire viewsphere is built from a database of synthetic 3D object models by deriving part structures for each class from the rendered training images such that intra-class and viewpoint variation is covered. The part structure allows for a 2D localization and an approximate 3D pose estimation on unseen test images.

In the present paper, we propose to reconsider previous work on unsupervised selection of informative features [4, 17, 24, 25] for the task of multi-view object class detection and pose estimation. We extend previous feature selection methods which were based on simple image patches [4, 24, 25], generative learning [15] or multi-level hierarchical models [26]. Instead, we rely on a discriminative learning of object parts; unlike [16], these parts are initialized in an unsupervised clustering step. Our approach is based on a flat hierarchy which can be efficiently evaluated using well-known spatial layout models [8]. We exploit the advantages of CAD models for training, as previously identified in [20, 21, 27], in order to perform a fully unsupervised selection of photometric object parts over the entire viewsphere. By avoiding manually chosen semantic part correspondences [27] or view-based subdivisions [20, 21], common geometry and appearance, which are intrinsic to each object class over the entire viewsphere are discovered without requiring any data set specific positive training examples. Consequently, an object part can contribute to the object class representation for different points on the

*acknowledge support by BMBF grant SiVe FKZ 13N10027

bike	bus	car	cat	cow	dog	horse	mbike	person	sheep
831	1455	17134	733	362	894	1161	370	1801	123

Table 1. Number of available pre-built synthetic 3D models from turbosquid.com for the PASCAL VOC2006 object classes.

viewsphere; see Figure 2. We demonstrate that our object class representation is suitable for object classes with significantly different appearance and geometry; it can be adapted to 2D detection and, in further contrast to [16], allows to infer an approximate 3D pose from the constellation of the shared parts. Despite a significantly leaner part representation it performs on par with or better than state-of-the-art on several test sets.

2. Object Class Representation

In this section, we describe each stage of building a representation of an object class which covers viewpoint and intra-class variation without requiring manual intervention.

2.1. Training Data

The approach derives its positive training examples for all subsequent steps entirely from a database of textured 3D CAD models (see Figure 1) and it draws all negative training examples from the VOC2006 data set. Such an object class specific database is established by downloading 3D models from distributors such as turbosquid.com or doschdesign.com and converting these models into a suitable format. Table 1 shows that a sufficient number of these 3D models is available for each of the VOC2006 object categories. In relying on 3D CAD models, our approach does not need to be retrained or adapted to the data set characteristics. This is a key advantage in reducing training set dependencies in favor of a better generalization. Each model in the database is rendered from a dense grid of points defined on the entire viewsphere in steps of $5°$ over azimuth α and elevation ϵ; further details are given in Section 4. The rendering of the models is done once in front of a black background, which we term the *pure training images*, and once in front of randomly selected images from the negative set, which we term the *validation images*. Lighting conditions are randomly varied in each rendering step. Training is performed on a single scale; yet, at test time, the approach is capable of detecting object classes on multiple scales.

2.2. Generating a Pool of Parts

Initially, for each object class, a pool P of object parts is generated as input for the subsequent higher-level training steps as follows: On each pure training image, HOG features [3] of different layout sizes are computed densely. Affinity propagation [9], which is an unsupervised clustering procedure, is applied to all features of each HOG layout collected from the pure training images. For certain object classes, it can be advantageous to enforce a balanced coverage of all viewpoints. This can be achieved by applying the clustering procedure repeatedly on subsets of features from subspaces on the entire viewsphere, instead of a single set of features on the entire viewsphere; see Section 4.1 for an evaluation of different balancing strategies. The features assigned to each cluster serve as the positive training examples for a linear SVM, trained against features from the negative training set. Each potential object part in the pool P is now represented by a linear SVM classifier.

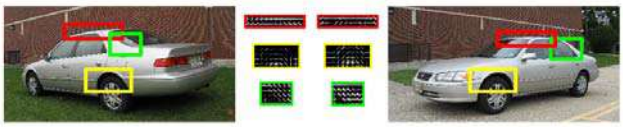

Figure 2. Object parts from different viewpoints may display similar appearance characteristics in HOG space. Our object class representation exploits these similarities in an adaptive way for detection and pose estimation.

2.3. Selecting the Most Informative Parts

Due to symmetries and self-similarity (see Figure 2), the pool P contains a large number of redundant or non-informative parts. In the next step, a subset of the object part pool is selected by ranking the informativeness of each part w.r.t. a positive and a negative image set with an entropy-based measure [25] and retaining only those parts which are most informative for the given task; altogether N parts are chosen until the informativeness of an additional part is below a threshold. Depending on the task setting, the informativeness of a part can be defined in different ways: for separation of an entire object class from the background, informative parts are those which appear on as many object instances under as many viewpoints as possible, whereas for precise pose estimation, informative parts are those which generalize over as many models as possible, but are visible under only a small range of viewpoints. In the first case, the negative set is chosen to contain negative training examples from the VOC2006 data set, whereas in the second case, all other pure training images are considered as the negative set.

2.4. Modeling a Dense Grid of Spatial Part Layouts

Each classifier associated with an object part is applied densely to each of the pure training images of a point on the viewsphere; its responses form part score maps as shown in Figure 5. However, each of the selected object part classifiers alone still does not offer sufficient discriminatory power for the task of detecting the presence of an object of a certain class and estimating its spatial extent. In a subsequent step, we build a generative model which describes the spatial occurrence layout of a small subset of M object parts ($M \subseteq N$) for each of the points on the entire viewsphere in order to provide an initial object location hy-

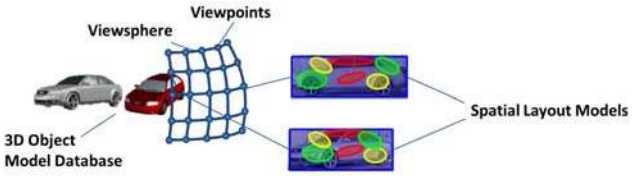

Figure 3. For each point on the densely sampled viewsphere, the spatial layout of informative parts is described by a spatial layout model.

pothesis (see Figure 3). In choosing a mixture of Gaussian distributions where all part locations are conditioned on the object center (see Section 3.2), we obtain a flat hierarchy which can be efficiently evaluated at test time as suggested in [7]. On the validation set, we optimize the trade-off between minimizing the number of parts M represented in the generative model in favor of a lean description, and maximizing the recall of the model on the expected intra-class and intra-viewpoint variation; see Section 4 for experimental results.

2.5. Learning the Global Object Class Appearance

During testing, the spatial layout models for all defined points on the entire viewsphere will allow generating a set of hypotheses. However, depending on each spatial layout, these hypotheses can have different score ranges and varying aspect ratios. In order to rank them against each other in a consistent way, we resize the hypotheses generated on the validation images to the training scale, convert them into spatial pyramid representations [12] and train a nonlinear SVM with an intersection kernel [11]. In using spatial pyramids, we can impose a regularly spaced grid subdivision which is relative to the area covered by a hypothesis and thus independent of its aspect ratio and dimension. The spatial pyramids encode the part score maps of all N selected object parts within the area covered by a hypothesis. They provide a more fine-grained appearance description and allow to jointly describe an entire object class with a single classifier. The nonlinear classification step results in a significant gain in precision of up to 40% in our experiments when compared to the scores of the spatial layout models alone. Note, however, that the nonlinear classifier is expensive to evaluate; in using the efficient spatial layout models to preselect object hypotheses, we can limit the nonlinear classification to a few hundred evaluations per test image.

2.6. Pose Estimation

Unlike object class detection, pose estimation requires that object parts be specific to a small range of viewpoints instead of covering the entire viewsphere. In adapting the selection criteria in Section 2.3 accordingly, we can draw a new subset of parts from the original pool P which satisfy this criterion for each of a set of suitably discretized subspaces of the viewsphere; note that the discretization can be

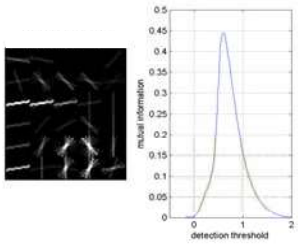

Figure 4. Mutual information as a function of varying the detection threshold of an example object part.

freely adapted to the task setting and is not inherent to or imposed by our training procedure. The steps described in Section 2.5 can then be repeated in the same way, replacing the single spatial pyramid classifier for the entire class by classifiers for each subspace. Within the discrete subspace with the highest classification score, the pose estimation precision can be further refined by modeling its spatial layout of object parts as described in Section 3.3. This allows the most likely pose parameters to be inferred from the spatial layout of the detected parts at test time.

3. Learning Methodology

In the following, we provide details on the learning methods used to build the object class representation.

3.1. Part Selection

In order to choose parts which are informative for detection or pose estimation, the first step of our entropy-based selection process is to determine the optimal detection threshold θ by maximizing the mutual information [2] between each object part in the pool P and sets of positive and negative images. To this purpose, an indication function p_n of an object $part_n$ in association with a detection threshold θ_n is defined as binary variable

$$p_n(I, \theta_n) = \begin{cases} 1, \text{ if } s_{max}(I, part_n) \geq \theta_n \\ 0, \text{ otherwise} \end{cases} . \quad (1)$$

Here s_{max} is the maximum score of the object part classifier (i.e. the linear SVM) in an image I. In addition, a binary class variable C is defined where $C(I) = 1$ if the image I belongs to the positive set of images and 0 otherwise. Between these two binary variables the mutual information $MI(p_n(\theta_n); C)$ is defined as

$$MI(p_n(\theta_n); C) = H(C) - H(C|p_n(\theta_n)) \quad (2)$$

with $H(x)$[1] and $H(x|y)$[2] being the marginal and the conditional entropy. As shown in Equation 2, the mutual information of an object part with its indication function p_n

[1] $H(x) = -\sum_x p(x) \log(p(x))$
[2] $H(x|y) = -\sum_{x,y} p(x,y) \log(p(x|y))$

depends on the detection threshold θ_n. Consequently, the optimal detection threshold θ_n^{opt} for an object part can be determined from

$$\theta_n^{opt} = \underset{\theta_n}{\operatorname{argmax}}[MI(p_n(\theta_n); C)]. \quad (3)$$

An example for the mutual information of an object part as a function of the detection threshold is given in Figure 4. If the detection threshold is set too low, the mutual information score will also be low since the object part is detected frequently in the negative images. A high detection threshold will likewise result in a low mutual information since the object part is now too sparsely detected in the positive images. At some intermediate value of the detection threshold the mutual information reaches a maximum and the object part delivers a maximum amount of information about the set of positive images.

After the optimal detection threshold for each object part in the pool is determined we can select an optimal subset of N parts from the pool P iteratively. Formally, this selection process can be described as

$$p_i = \underset{p_m \in P_i}{\operatorname{argmax}}[\min_{p_l \in S_i}[MI(p_l, p_m; C) - MI(p_l; C)]]. \quad (4)$$

Here P_i is the pool of available object parts at iteration i and S_i is the subset of selected object parts from the pool at iteration i. The minimum taken over all previously selected parts p_l avoids redundancy. The maximum is taken over all object parts p_m from the pool P_i to ensure that a part is selected that yields the maximum increase of information about the set of positive images. The update rules for the pool of available object parts and the subset of selected object parts are defined by

$$P_{i+1} = P_i \setminus p_i \quad S_{i+1} = S_i \cup p_i \quad 1 \leq i \leq N+1. \quad (5)$$

The initial pool P_1 contains all generated object parts from P and the initial subset of selected parts S_1 is an empty set. After $N+1$ iterations we have selected N object parts from the pool that contain a maximum of information regarding the sets of positive and negative images.

3.2. Spatial Layout Model

To jointly describe the spatial occurrence and the detection uncertainty of the informative object parts, a spatial layout model is built for each point on the entire viewsphere. Each classifier associated with an object part is applied densely to each of the pure training images of a point on the viewsphere; its responses form part score maps as shown in Figure 5. The location of the maximum classifier score in each score map is stored, resulting in a set of 2D locations for each part. For each object part we fit a mixture of Gaussian distributions [1] to the locations to obtain one spatial layout model for each densely sampled viewpoint, conditioned on the location of the center of the object and linked to the mean training bounding box size (see Figure 5). During testing, for an unseen object, its bounding box can be inferred by evaluating the probability of the location of each detected part w.r.t. the spatial layout model; in practice, this evaluation can be efficiently performed using distance transforms [7]. The number of informative object parts to be considered in the spatial layout model is linked to the trade-off between a high recall and a low model complexity. In practice, we determine the optimal number of object parts in a validation step; see Section 4 for an experimental evaluation.

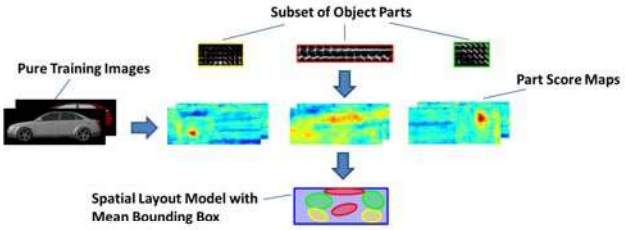

Figure 5. For each point on the entire viewsphere we resize the training images to the corresponding mean bounding box and apply the classifiers corresponding to the respective subset of object parts. We model the occurrence of these parts by using a mixture of Gaussians distributions to describe the spatial distribution of their detection scores.

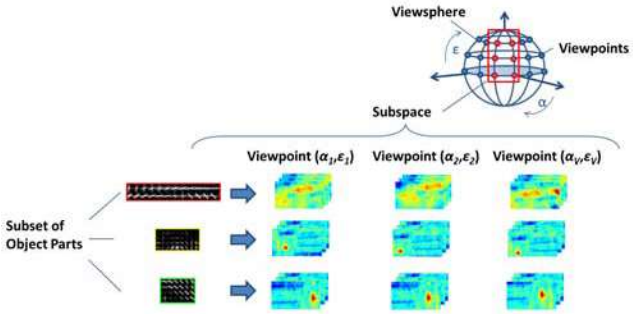

Figure 6. The pose refinement step is based on several Gaussian mixture models. Each Gaussian mixture model captures the spatial arrangement of an object part for a point on the viewsphere.

3.3. Pose Refinement

As described in Section 2.6, the pose estimation is based on several classifiers where each classifier is directly linked to a discretized subspace of the viewsphere. Figure 6 shows how the estimation precision can be further refined for the subspace with the highest score of the corresponding spatial pyramid classifier (see Section 2.6). Each of the discretized subspaces has its informative subset of N object parts and consists of several viewpoints $\omega_v = \{\alpha_v, \epsilon_v\}$. During training, the subset of object classifiers is applied densely to each of the pure training images within the corresponding subspace. As described in Section 3.2, for each object part we again fit a mixture of Gaussians to the locations of the maximum classifier score to obtain a spatial layout model for each object part and each viewpoint within the subspace. Assuming that such a model has K components, the proba-

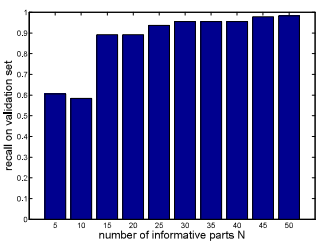

Figure 7. Tradeoff between recall on the validation set and number of selected parts for spatial layout models of the bicycle class.

bility p of an object $part$ for a viewpoint ω is given by

$$p(part|\omega) = \sum_{k=1}^{K} p_k(part|\omega). \quad (6)$$

Assuming conditional independence of the N parts, the estimated pose can be refined to the viewpoint

$$\omega^{est} = \underset{\omega}{\operatorname{argmax}} \prod_{n=1}^{N} p(part|\omega) = \prod_{n=1}^{N} \sum_{k=1}^{K} p_k(part|\omega). \quad (7)$$

Note that the refined viewpoint in Equation 7 is relative to the virtual camera parameters used to generate training images from our synthetic 3D model database. With an estimated viewpoint and a bounding box of the 2D localization step we are able to project a mean 3D bounding box computed from the 3D model database into each tested image. Some examples are shown in Figure 12.

4. Experimental Results

In this section we outline the results we achieve with our proposed model for object localization and pose estimation. For these two tasks we evaluate the performance of our approach for cars and bicycles on the 3D Object Category data set introduced by [19]. It is the current state-of-the-art data set for multi-view object class detection and pose estimation. For each object class the 3D Object Category data set contains 10 different object instances; for each instance 48 images are provided. They are taken at 8 different azimuth angles in 45° steps *(back, back-left, left, front-left, front, front-right, right, back-right)* for 2 different elevation angles, and 3 different scales. In order to evaluate the 2D localization we use the overlap criterion suggested by [6]: a predicted bounding box is considered as correct if the overlap between a predicted bounding box and a ground truth bounding box exceeds 50%. If several bounding boxes are predicted in the same image area, only the highest scoring detection is considered as correct, while the remaining detections are considered as false positives.

Our approach relies exclusively on training data rendered from pre-built 3D models, which are available from the distributors turbosquid.com and doschdesign.com. For training, we use 25 car models and 8 bicycle models. In order

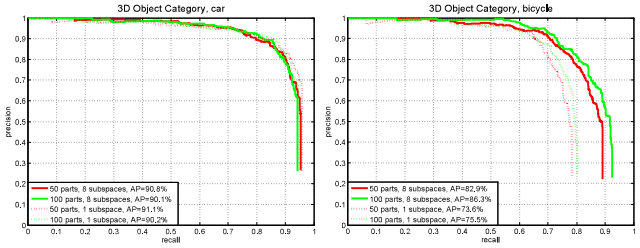

Figure 8. Precision/Recall curves for the 3D Object Category data set car (left) and bicycle (right) using different viewsphere subspaces to balance the part generation (solid/dashed), and different maximum numbers of object parts for part selection (red/green).

to define a dense grid of viewpoints on the viewsphere, azimuth is sampled from 0° to 360° in 5° steps and elevation is sampled from 0° to 20° in 5° steps. This viewpoint setup is used to generate the pure training images and the validation set. We draw all negative training images from the PASCAL VOC2006 training data set excluding the training images for the object classes car and bicycle. For our experiments we rely on the HOG descriptor of [8] with a HOG cell size of 8 pixels; the spatial encoding is done with a spatial pyramid on three levels of linear subdivision.

In Section 2.4 we outline that the number of selected object parts for the spatial layout models is chosen in a validation step to optimize the tradeoff between recall and model complexity. Figure 7 shows the impact of changing the part number w.r.t. the achieved recall for the object class bicycle. In this example, saturation is reached when selecting 50 object parts.

4.1. Subspaces and Number of Object Parts

In the first experiment, we evaluate the impact of two parameters of our learning procedure for the 2D localization task: the influence of balancing the part generation over the viewsphere, and the influence of the maximum number of selected object parts. For both classes, cars and bicycles, we compare the 2D detection performance on the 3D Object Category data set when generating informative parts from the entire viewsphere or when balancing the part generation to equally cover eight viewsphere subspaces. For each setup of both classes, we learn an object class representation with 50 and 100 selected parts. We apply our approach to the entire 3D Object Category data set containing all 480 test images per class. The precision/recall curves we obtain are shown in Figure 8. For cars, the number of subspaces has little effect on the overall detection accuracy, the reason being that for both settings, suitable object parts for each subspace are generated in a balanced way and the resulting object class detector does not suffer from low recall. In addition, the number of selected object parts has little effect on the performance of the car detector. For bicycles the behaviour is different, since the front and back views which cover relatively small areas and contain delicate structures,

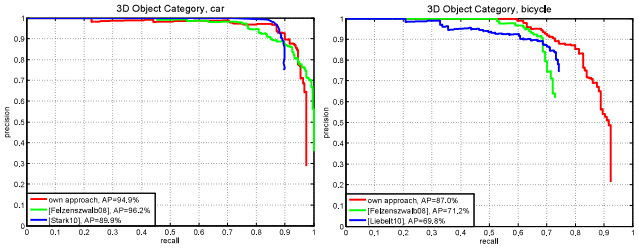

Figure 9. Precision/Recall curves for the 3D Object Category data sets car (left) and bicycle (right) for our approach (red curves) compared to state-of-the-art detectors.

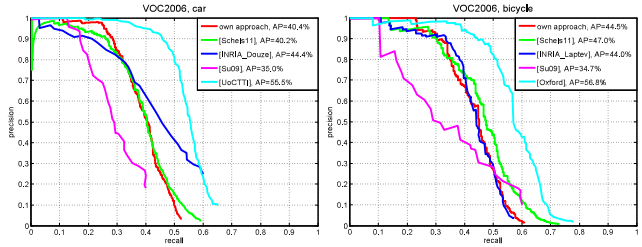

Figure 10. Precision/Recall curves for the PASCAL VOC2006 car (left) and bicycle (right) data set of our approach (red curves) compared to state-of-the-art detectors.

contribute fewer parts if the generation is performed on a single viewsphere; in choosing eight subspaces, the generated informative object parts are more equally distributed over the viewsphere, resulting in a higher recall. Furthermore, increasing the maximum number of selected object parts from 50 to 100 parts improves the overall precision of the bicycle detector.

Based on these results, for the subsequent tests 8 subspaces and at most 50 parts are chosen for the car detector and 8 subspaces and at most 100 parts for the bicycle detector. For training the 3D pose estimation, we also rely on the 8 subspaces and for each subspace we select a subset with at most 50 object parts for cars and at most 100 object parts for bicycles.

4.2. Object Localization

In order to compare our detection approach on the 3D Object Category data set to previous work, we follow the test protocol of [21] for cars and the test protocol of [13] for bicycles. Note that these test protocols define test subsets and therefore differ from the test setup for the experiments in Section 4.1 which are evaluated on the entire data set. For both classes, we also compare against the current state-of-the-art approach of [8] using their pre-trained object class models provided as part of *voc-release3*. As shown in Figure 9, with 94.9% on the car data set our approach can compete with the detector of [8] (96.2%) and outperforms the approach of [21] (89.9%), despite our detector being trained on different, i.e. synthetically generated, training images. Note that the approach of [21] which is also trained on synthetic data uses a bank of 36 viewpoint-specific models with more than 400 trained object parts. In contrast, our detector which is able to exploit appearance co-occurrences across different viewpoints requires only 50 object parts. With 87.0% on the bicycle data set we outperform the approaches of [13] (69.8%) and [8] (71.2%) due to a significantly higher recall.

We also evaluate our approach regarding 2D localization on the publicly available PASCAL VOC2006 [6] data set for cars and bicycles. The precision/recall curves we obtain with our proposed approach are given in Figure 10 (red curves). For both object classes we provide the best performing approaches of the PASCAL challenge 2006 [6] (blue curves), the best performing approaches of the PASCAL challenge 2007 on the 2006 test set [5] (cyan curves) and the most recent multi-view approaches of [20] (green curves) and [22] (magenta curves). Although we train our detector on a synthetically generated training set, our detection results (40.4% on cars and 44.5% on bicycles) can compete with these state-of-the-art detectors. Compared to the 3D Object Category data set, the lower recall of our approach on the VOC2006 data set may be due to the more pronounced object appearance variations within VOC2006 which are not sufficiently covered by the selected 3D training models.

4.3. Pose Estimation

In order to benchmark the 3D pose estimation performance of our approach on the 3D Object Category data set, we bin the estimated viewpoints of the pose refinement step (see Section 3.3) in $45°$ steps to match to the groundtruth annotations of [19]. Here we follow the test protocol of [21] for cars and the protocol of [13] for bicycles in order to compare our 3D pose estimation approach to existing approaches. The confusion matrices obtained by classifying all positive detections are shown in Figure 11. For cars we observe that confusion is more pronounced for opposing views due to the symmetries inherent in the car class. Still, the average accuracy of 82.6% compares favorably to the reported result of [21] (80.5%). For bicycles we observe that confusion is more pronounced between neighboring viewpoints. On the bicycle data set, the achieved result of 87.7% significantly outperforms [13] (75.0%). Figure 12 shows some results of the full detection process with 2D localizations and 3D pose estimations. Pose estimation typically fails when there is ambiguous or insufficient evidence in the image for a correct pose initialization; see two examples with red outlines in Figure 12.

Numerous approaches have evaluated on the 3D Object Category data set. However, different test configurations have been used, which makes an objective and comprehensive benchmarking difficult. To compare to each approach, we evaluate our approach using each of the test configurations reported by the different authors on the car data set.

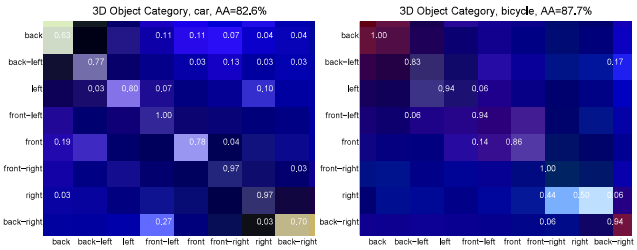

Figure 11. Confusion matrices (rows: groundtruth, columns: estimates) for the 3D Object Category data sets car (left) and bicycle (right).

Approach	Reported Test Configuration	AP_{2D}	Own AP_{2D}	AA_{3D}	Own AA_{3D}
Glasner [10]	5 inst./3 sc.	**99.2%**	94.9%	**84.9%**	82.6%
Liebelt [13]	3 inst./3 sc.	76.7%	**97.2%**	70.0%	**81.5%**
Schels [20]	10 inst./3 sc.	82.0%	**90.8%**	62.6%	**82.2%**
Stark [21]	5 inst./3 sc.	89.9%	**94.9%**	80.5%	**82.6%**
Su [22]	5 inst./2 sc.	55.3%	**94.9%**	≈69.4%	**83.6%**
Sun [23]	5 inst./2 sc.	—	**94.9%**	66.6%	**83.6%**
Zia [27]	5 inst./3 sc.	90.4%	**94.9%**	**84.0%**	82.6%

Table 2. We evaluate our approach following the previously reported test protocols on the 3D Object Category data set for cars in order to achieve an objective comparison. (abbr.: inst.=object instance, sc.=scale, AP_{2D}=average precision for 2D localization, AA_{3D}=average accuracy for 3D pose estimation).

The results are shown in Table 2. Note that our approach performs on par with or better than most of these state-of-the-art detectors for both 2D detection and 3D pose estimation, despite being trained synthetically.

5. Conclusion

We have presented an approach which learns an object class representation from a database of 3D CAD models without requiring any manual supervision. It exploits appearance co-occurrences due to symmetries and self-similarity by choosing non-semantic parts in a flexible framework suitable for 2D localization and 3D pose estimation. We demonstrate state-of-the-art performance on several test sets without having to use data set specific positive training examples. Future work will focus on the unsupervised learning of a deformable 3D spatial layout in order to combine object class detection and instance identification.

References

[1] C. A. Bouman. Cluster: An unsupervised algorithm for modeling Gaussian mixtures. Available from http://www.ece.purdue.edu/~bouman, 1997.

[2] T. Cover and J. Thomas. *Elements of Information Theory*. J. Wiley, 1991.

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[4] B. Epshtein and S. Ullman. Feature hierarchies for object classification. In *ICCV*, 2005.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results, 2007.

[6] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results, 2006.

[7] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 2005.

[8] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[9] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 2007.

[10] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011.

[11] K. Grauman and T. Darrell. The pyramid match kernel: Efficient learning with sets of features. *JMLR*, 2007.

[12] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[13] J. Liebelt and C. Schmid. Multi-view object class detection with a 3D geometric model. In *CVPR*, 2010.

[14] L. Mei, J. Liu, A. Hero, and S. Savarese. Robust object pose estimation via statistical manifold modeling. In *ICCV*, 2011.

[15] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006.

[16] P. Ott and M. Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011.

[17] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009.

[18] N. Payet and S. Todorovic. From contours to 3D object detection and pose estimation. In *ICCV*, 2011.

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

[20] J. Schels, J. Liebelt, K. Schertler, and R. Lienhart. Synthetically trained multi-view object class and viewpoint detection for advanced image retrieval. In *ICMR*, 2011.

[21] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3D CAD data. In *BMVC*, 2010.

[22] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009.

[23] M. Sun, H. Su, S. Savarese, and L. Fei-Fei. A multi-view probabilistic model for 3D object classes. In *CVPR*, 2009.

[24] A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing visual features for multiclass and multiview object detection. *PAMI*, 2007.

[25] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *ICCV*, 2003.

[26] L. Zhu, Y. Chen, A. Torralba, W. Freeman, and A. Yuille. Part and appearance sharing: Recursive compositional models for multi-view multi-object detection. In *CVPR*, 2010.

[27] Z. Zia, M. Stark, K. Schindler, and B. Schiele. Revisiting 3D geometric models for accurate object shape and pose. In *ICCV 3dRR-11 Workshop*, 2011.

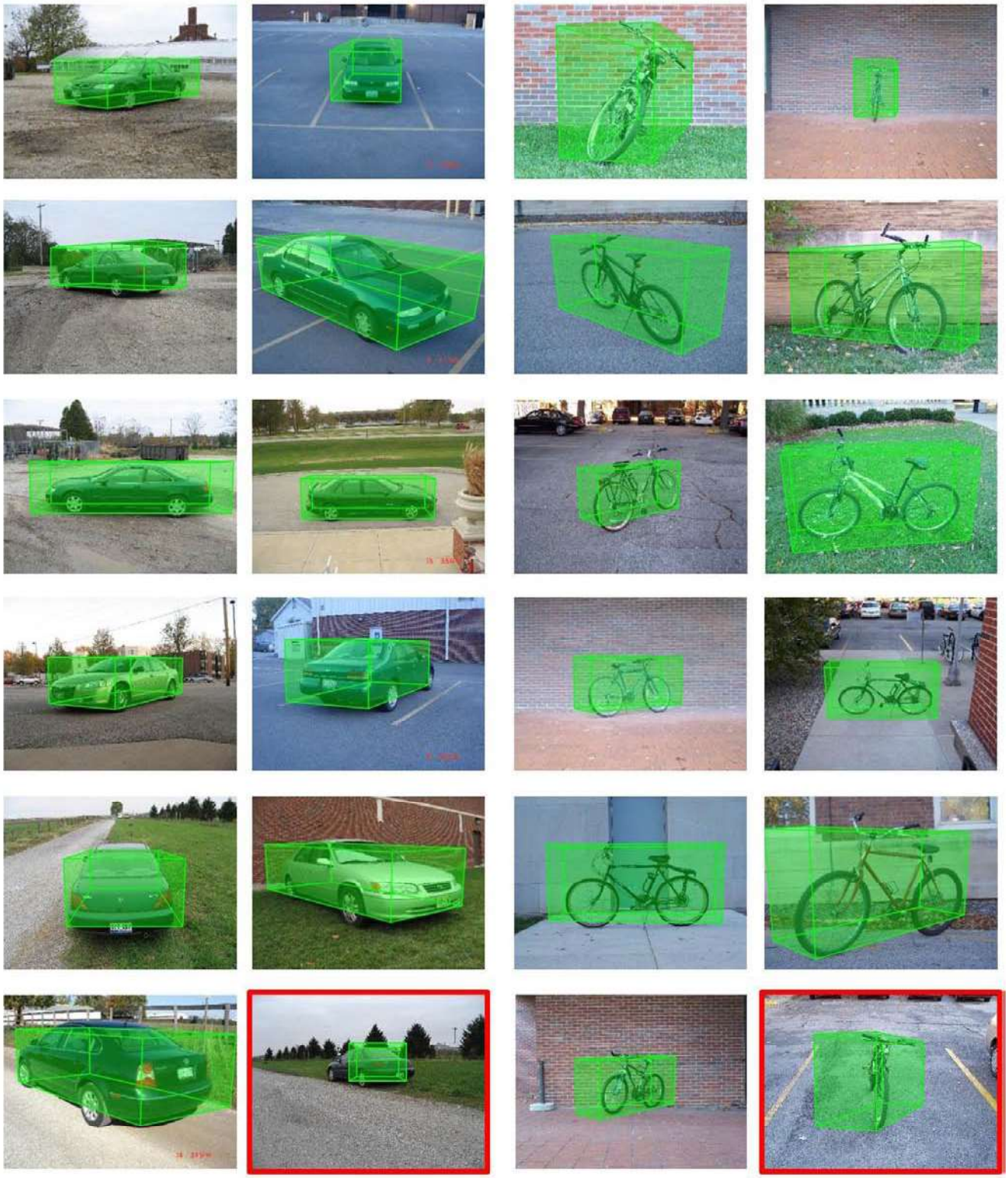

Figure 12. Some detection results of our proposed approach on the 3D Object Category data sets car (left) and bicycle (right). For both object classes an incorrect detection example is shown (indicated with a red outline). This figure is best viewed in color.