Back to the Future: Learning Shape Models from 3D CAD Data

Michael Stark[1]
stark@mpi-inf.mpg.de
Michael Goesele[2]
goesele@cs.tu-darmstadt.de
Bernt Schiele[1]
schiele@mpi-inf.mpg.de

[1] MPI Informatics,
Saarbrücken, Germany

[2] Computer Science Department,
TU Darmstadt, Germany

Abstract

Recognizing 3D objects from arbitrary view points is one of the most fundamental problems in computer vision. A major challenge lies in the transition between the 3D geometry of objects and 2D representations that can be robustly matched to natural images. Most approaches thus rely on 2D natural images either as the sole source of training data for building an implicit 3D representation, or by enriching 3D models with natural image features. In this paper, we go back to the ideas from the early days of computer vision, by using 3D object models as the only source of information for building a multi-view object class detector. In particular, we use these models for learning 2D shape that can be robustly matched to 2D natural images. Our experiments confirm the validity of our approach, which outperforms current state-of-the-art techniques on a multi-view detection data set.

1 Introduction

In the 70's and 80's the predominant approach to recognition was based on 3D representations of object classes [4, 17, 18, 19, 22]. While being an intriguing paradigm these approaches showed only limited success when applied to real-world images. This was due to both the difficulty to robustly extract 2D image features as well as their inherent ambiguity when matching them to 3D models. Today, thirty years later, the predominant paradigm to recognition relies on robust features such as SIFT [16] and powerful machine learning techniques. While enabling impressive results, e.g., for the PASCAL-VOC challenge [6], these methods have at least two inherent limitations. First, methods typically do not allow to recognize objects from arbitrary viewpoints but are limited to single viewpoints instead. And second, these approaches rely on the existence of representative and sufficient real-world image training data for object classes limiting their generality and scalability.

The starting-point of this paper is therefore to go back to the idea of using 3D object models only and re-examine the problem of object class recognition from such 3D data alone, not using any natural training images of the object class. In contrast to early approaches, we draw from a multitude of advancements in both object class recognition and 3D modeling, which we use as tools for designing highly performant object class models. The first and

BMVC 2010 doi:10.5244/C.24.106

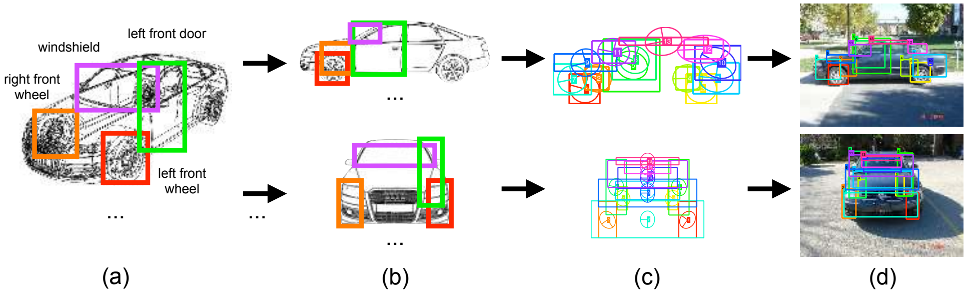


Figure 1: Learning shape models from 3D CAD data. (a) Collection of 3D CAD models composed of semantic parts, (b) viewpoint-dependent, non-photorealistic renderings, (c) learned spatial part layouts, (d) multi-view detection results.

most important tool is an abstract shape representation that establishes the link between 3D models and natural images, based on non-photorealistic rendering. The second tool is a collection of discriminatively trained part detectors, based on robust dense shape feature descriptors on top of this representation. The third tool is a powerful probabilistic model governing the spatial layout of object parts, capable of representing the full covariance matrix of all part locations. All three tools aim at capturing representative object class statistics from a collection of 3D models, increasing the robustness of the resulting object class models.

The main contributions of our paper are as follows. First, we revisit the problem of object class recognition entirely based on 3D object models, avoiding the need for any natural training images of the objects. Second, we propose an abstract shape representation in connection with robust part detectors that establishes the link between 3D data and natural images. Third, we evaluate our model in a series of experiments with respect to multi-view detection and viewpoint classification (pose estimation), and demonstrate superior performance compared to state-of-the-art techniques on a standard multi-view recognition benchmark.

2 Related work

Recognition of 3D objects has a long history. While many of today's approaches model single 2D views rather than 3D objects, 3D object class models have been revived recently as object recognition is inherently related to the object's three dimensional nature. 3D object class models are typically built either implicitly, by organizing training images according to their position on the viewing sphere, or explicitly, by establishing correspondences between training images and a given 3D geometry representative for an object class. In both cases, and in addition to representing 3D constraints, the robust encoding of object appearance learned from a sufficient amount of natural training images is considered key to success.

The first major line of research in 3D object recognition starts from a collection of natural training images depicting the object class of interest from varying viewpoints [2, 11, 21, 26, 27]. The viewpoint itself is treated either as an observed [11, 14] or unobserved [26] variable, resulting in different amounts of supervision needed during training. Establishing correspondences between image features from different views by means of tracking [27] or imposing affine transformations [26] can then be used as the basis for rough estimates of three dimensional object geometry. These approaches have adopted sophisticated techniques to compensate for the large amount of required training data, such as sharing information between multiple codebooks by activation links [27], similarity transforms [2], or by synthesizing unseen viewpoints by means of a morphing variable [26]. However, due to the reliance on sufficient training data from multiple viewpoints they are still bound to a typically coarse 3D and viewpoint representation of the object class, limiting the amount of variation captured by both appearance and geometry representations.

The second major line of research thus starts from a given 3D geometry representative for an object class, typically given in the form of one or a few 3D models [14, 29], which is assumed to capture geometric variation better than a model built from a limited collection of viewpoint images. The geometry model then serves as a reference frame to which supplemental natural training image features are attached, which can then be matched to natural images for recognition. While [29] performs the attachment based on appearance similarity, [14] establishes the link between images and geometric model by spatial consistency. In particular, the geometric model is rendered from the same viewpoints as the training images (requiring viewpoint annotations). Both are overlaid the same regular grid, establishing correspondences between respective grid positions. However, these approaches still require a sufficient number of supplemental real-world training images again limiting their generality.

Rather than using real-world training images we go back to the idea of early papers [4, 17, 18, 19, 22] to use 3D object models alone. More specifically we resort to using 3D computer aided design (CAD) models exclusively, both for learning local shape and global geometry models for the object class of interest. Abandoning object training images altogether additionally circumvents the need for an attachment step, which is susceptible to introducing noise into the appearance representation. We note that [15] is also exclusively based on 3D CAD data, but has been superseded by [14], which in turn is outperformed by our approach (see Sect. 5). Our work is different from [15], in that we explicitly design an abstract shape representation for 3D CAD data that can be directly matched to natural images, while [15] uses photorealistic rendering techniques against varying backgrounds to produce features resembling the ones found in natural images. We further suggest a full covariance spatial model for capturing the geometric variation of a collection of 3D CAD models, while [15] resorts to a star model (via generalized Hough voting).

3 Object class representation

Our object class representation combines two prominent approaches. First, it represents object classes as an assembly of spatially arranged parts, which has been shown to be an effective strategy for dealing with intra-class variation and partial occlusion for generic object class recognition [7, 13]. Second, it subsumes object classes in a collection of distinct models, where each model corresponds to a discrete viewpoint. For each viewpoint, the link between 3D CAD models used for training and natural test images is established by a local shape representation of object parts, based on non-photorealistic rendering.

3.1 Object classes as flexible part configurations

In the spirit of [8], we choose a part-based object class representation as the basis for our approach. Instances of a given object class are assumed to consist of a fixed set of parts, subject to both constraints describing their spatial layout and their relative sizes. Following early uses of CAD models for recognition [4], but in contrast to recent work [14, 15], we chose

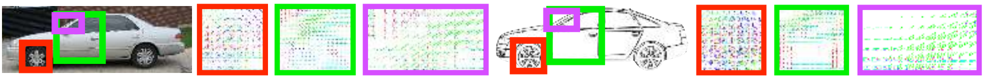


Figure 2: Comparison of shape representations fed into Shape Context descriptors for a real image (left) and a rendered 3D CAD model (right). For each colored bounding box, we show overlapping edge image patches, where edge orientation is encoded as hue and edge strength as saturation. Best viewed in color with magnification.

to not only use the three dimensional geometry from 3D CAD data, but additionally exploit included semantic information. In particular, we benefit from the fact that CAD data is typically created by human designers, often following intuitive routes when building complex models from simpler building blocks. As an example, consider the car model of Fig. 1 (a), which has been composed of semantically meaningful parts, such as wheels, doors, a roof, etc. While we cannot expect arbitrary 3D CAD models from the web to offer consistent part decomposition and labeling, we observe that all 41 car CAD models in our data base[1] share a common set of approximately 20 parts, from which we use 13 in our experiments (four wheels, both front doors, both bumpers, hood and trunk, windshield and rear window, the roof; see Fig. 1 (c)). Since both part decomposition and naming are potentially preserved in modern CAD file formats, we can establish semantic part-level correspondences between CAD models with minimal labeling effort. Inferring candidate parts and their correspondences automatically based on 3D geometry would be an alternative [24].

3.2 Viewpoint-dependent shape representation

In order to map 3D CAD data parts to the image plane, we apply a perspective projection according to the viewpoint of interest. In the image plane, each part is characterized by an axis-aligned bounding box (see Fig. 1 (a,b)). Note that we can still identify a part with a bounding box even in case it is not visible due to object-level self occlusion, as is the case for the right front wheel in the left side view of the car of Fig. 1 (b). In this case, the contents of the bounding box (orange) will depict the occluder (portion of the left front wheel, left front fender), not the originating object part. Following parts through occlusion in this fashion has the advantage of rendering occlusion reasoning superfluous, simplifying the design of the model. Coherence of part shapes between neighboring viewpoints also falls out naturally.

In contrast to earlier attempts at learning appearance models from 3D CAD data [15], we choose a shape-based abstraction of object appearance at the core of our part-based representation. We focus on capturing edge information, which we expect to be repeatable across 3D CAD models of a given object class as well as natural images depicting instances of that class. At the same time, using the edge abstraction eliminates the need for rendering CAD models multiple times under varying lighting conditions, textures, and backgrounds, and having a learning algorithm finding out about relevant gradients afterwards. This intriguing property was shared by early approaches [4, 17, 18, 19, 22], but is often neglected by recent object class models. Specifically, we render three different types of edges for any 3D CAD model: crease edges, which are inherent properties of a 3D mesh, and thus invariant to the viewpoint, part boundaries, which mark the transition between object parts and often coincide with creases, and silhouette edges, which describe the viewpoint-dependent visible outline of an object [12]. In all three cases, we render edge strength (determined by the

[1] Commercial models from www.doschdesign.com

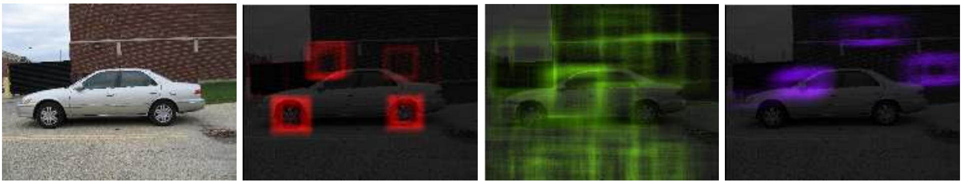


Figure 3: Part detector responses for *left front wheel* (red), *left front door* (green), and *windshield* (magenta), overlaid onto the original image (left). For each part, we show an accumulation of bounding boxes, weighted by detector response, drawn at the respective location and scale.

sharpness of the crease for crease edges) as well as orientation in the image plane.

In order to describe the contents of a part bounding box in the image plane, we use a specific flavor of Shape Context [3] descriptors that has proven to be highly robust in the context of object class detection in cluttered images [1]. These descriptors are densely sampled over a uniform grid of overlapping image patches, and accumulate edge orientations locally in log-polar histograms. Fig. 2 visualizes edge information fed into these descriptors for both a non-photorealistic rendering of a 3D CAD model (right) and a natural image (left). Please note the high degree of visual similarity between the two visualizations. It indicates that the chosen shape abstraction successfully captures common properties of both renderings and natural images, which we consider a key ingredient for robust recognition.

4 Multi-view object class detection

As outlined in Sect. 3, our multi-view object class detection framework is based on a set of distinct object class models, one for each particular viewpoint, which is sometimes referred to as a *bank of detectors* [27]. All models are structurally equal, the only difference between them is the viewpoint-dependent data used for training. Final detection hypotheses are generated by combining hypotheses from the individual models.

4.1 Discriminative part shape detectors

In order to discriminate between object parts and image background, we use the highly performant part shape detectors proposed by [1] in connection with the shape context features described in Sect. 3.2. For each object part, we train an Ada-Boost classifier [9] on positive and negative training examples. Positive examples are obtained via non-photorealistic rendering of the object part in question. Negative examples are randomly sampled from a background image set, not containing the object class of interest. The set of positive training examples is further artificially enhanced by adding slightly translated and scaled (jittered) copies of the original examples. During detection, the trained classifier is evaluated in a sliding-window fashion at different image positions and scales. Fig. 3 gives example part responses for three different car parts in a left side view. We transform Ada-Boost classifier responses into pseudo-likelihoods using Platt scaling [20], and form a set of discrete candidate part locations (typically up to several 100K per part and image) by applying a threshold.

4.2 Probabilistic spatial model

Interestingly, most recent work in multi-view recognition has adopted star-shaped spatial models [2, 15, 26, 27, 29]. In contrast to these prior works, our approach uses a more powerful probabilistic representation of the spatial layout of parts inspired by the constellation model [7]. We use an efficient implementation along the lines of [25], from which we borrow the notation in the following paragraphs. The probabilistic formulation combines the shape of individual parts S, their relative scales R, and their overall spatial layout X. During detection, the goal is to find an *assignment* of all P model parts to candidate part locations in a test image, denoted as the detection hypothesis $H = (h_1, \ldots, h_P)$. That is, h_p contains a candidate part identifier assigned to part p. The detection problem can be formulated as a maximum a posteriori (MAP) hypothesis search over the distribution $p(X,R,S,H|\theta)$, which is the joint posterior distribution of H and image evidence, given a learned model θ. It factors into separate likelihood contributions for local part shape, spatial part layout, relative part scales, and a (uniform) prior on hypotheses, as follows:

$$p(X,R,S,H|\theta) = \underbrace{p(S|H,\theta)}_{Local\,Shape}\ \underbrace{p(X|H,\theta)}_{Layout}\ \underbrace{p(R|H,\theta)}_{Relative\,Scale}\ \underbrace{p(H|\theta)}_{Prior} \qquad (1)$$

Local part shape. Local part shape $S(h_p)$ is modeled as a product of independent pseudo-likelihoods from Platt-scaled Ada-Boost classifier responses $p(S(h_p)|\theta)$.

$$p(S|H,\theta) = \prod_{p=1}^{P} p(S(h_p)|\theta) \qquad (2)$$

Spatial layout and relative scales. Spatial layout of parts is modeled as a joint Gaussian distribution over their coordinates $X(H)$ in a translation- and scale-invariant space (the *constellation*), using Procrustes analysis [5]. The model allocates independent Gaussians for the relative scale $R(h_p)$ of each part, i.e., the ratio between part and constellation scale.

$$p(X|H,\theta)\,p(R|H,\theta) = \mathcal{N}(X(H)|\theta) \prod_{p=1}^{P} \mathcal{N}(R(h_p)|\theta) \qquad (3)$$

Learning and inference. Since we assume the densities for relative scales and spatial layout to be Gaussian, we can estimate parameters θ in a maximum likelihood fashion, given part-level correspondences. Following [25], we use an efficient Data-Driven Markov Chain Monte Carlo sampling algorithm for MAP inference. We approximate the MAP hypothesis $H_{\mathrm{MAP}} = \arg\max_H p(H|X,R,S,\theta)$, which is is equivalent to $\arg\max_H p(X,R,S,H|\theta)$, by drawing samples from $p(X,R,S,H|\theta)$ using the Metropolis-Hastings (MH) algorithm [10]. Employing the single component update variant of MH allows to separately update individual components of the target density, conditioned on the remaining portion of the current state of the Markov chain. This opens the possibility to guide the sampling towards high density regions by data-driven, bottom-up proposals [28, 30], which we instantiate by part shape likelihoods $p(S(h_p)|\theta)$.

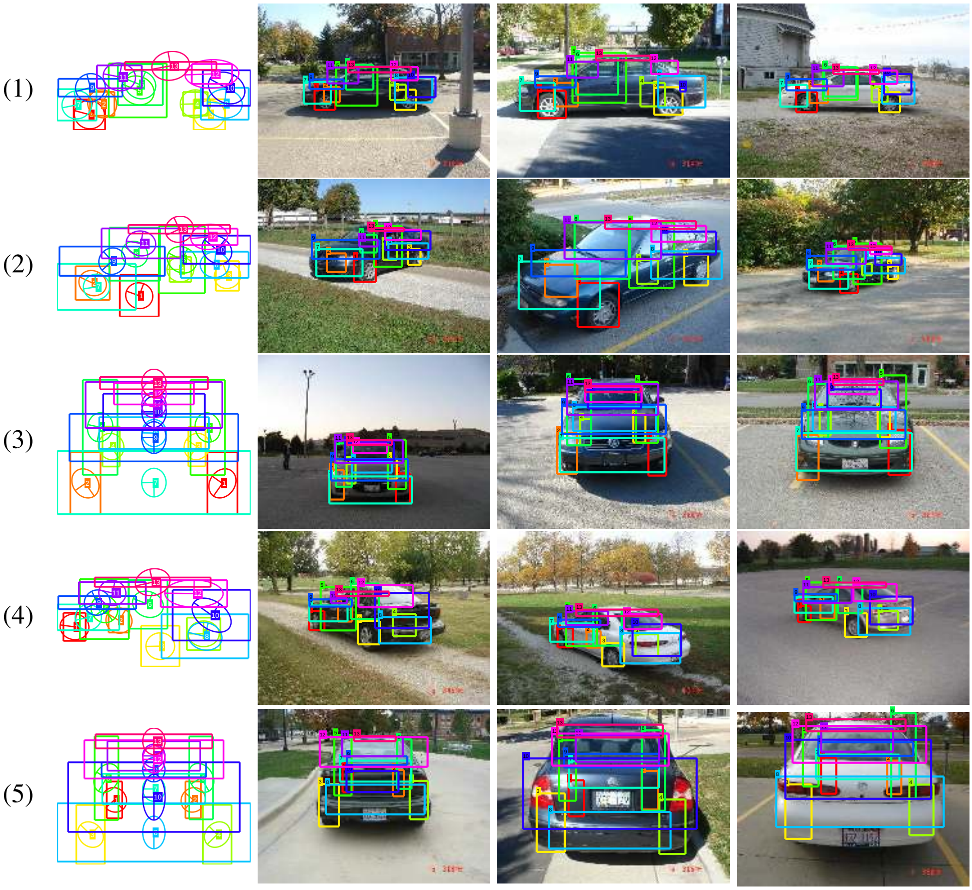


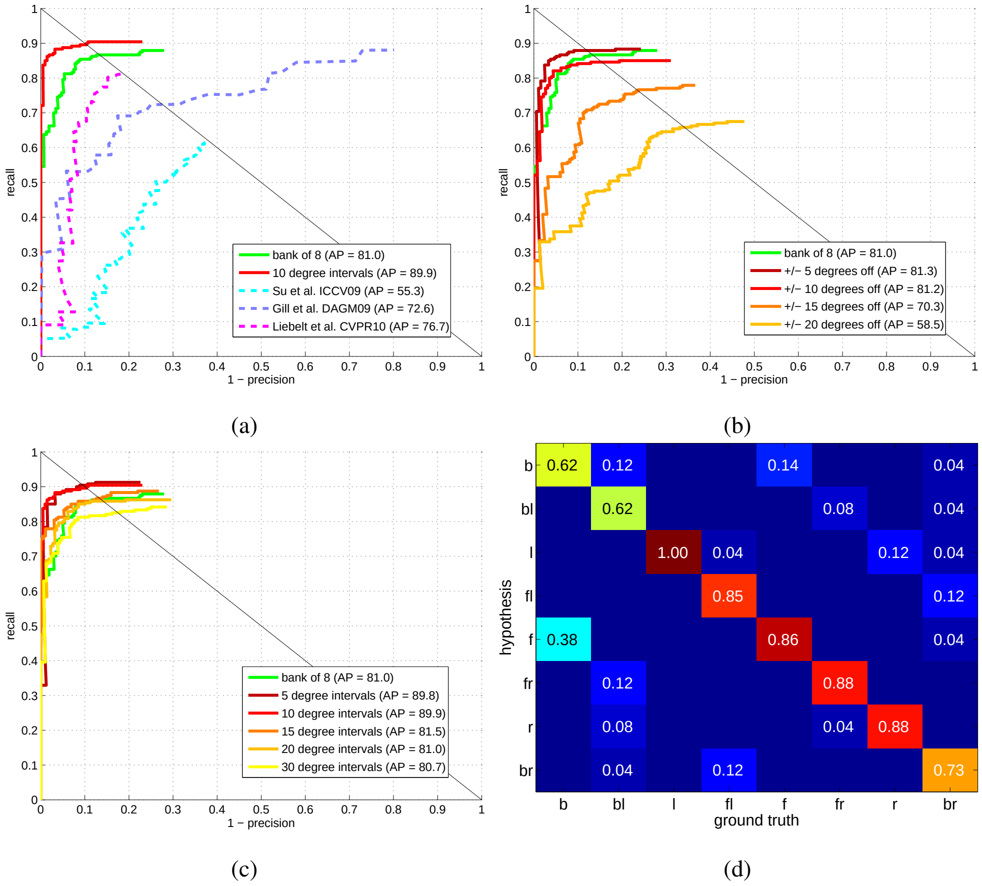
Figure 4: Viewpoint-dependent object class models for the viewpoints *left* (1), *front-left* (2), *front* (3), *back-left* (4), and *back* (5) (left-most column). Ellipses denote positional variance of parts, which are drawn at the learned mean scales. Example detections (right columns).

4.3 Viewpoint estimation

In order to be able to detect potentially multiple object instances in an image, we run a number of independent Markov chains (typically 50) for each viewpoint-dependent detector of a bank. For each chain, we memorize the highest-scoring bounding box together with the viewpoint of the originating detector. We then apply a standard, greedy, overlap-based non-maximum suppression on all bounding box-viewpoint pairs, and retain all survivors as the final hypotheses concerning object bounding box and viewpoint.

5 Experimental evaluation

We evaluate the performance of our model on the *car* class of the 3D Object Classes data set introduced by [23]. The data set has been explicitly designed as a multi-view detection benchmark, containing 10 different cars, each pictured in front of varying backgrounds from



(a)

(b)

(c)

(d)

Figure 5: Multi-view object class detection results, (a) comparison to state-of-the-art ([11, 14, 26]), (b) varying amounts of perturbation w.r.t. the true annotated viewpoint, (c) varying densities of sampled viewpoints, (d) confusion matrix for viewpoint classification.

8 different 45 degree-spaced azimuth angles (*left*, *front-left*, *front*, *front-right*, *right*, *back-right*, *back*, *back-left*), 2 different elevation angles (*low*, *high*), and 3 different distances (*close*, *medium*, *far*). The resulting 48 viewpoints are typically not fully accurately met, but may be off by a few degrees in either direction. We evaluate object class detection from multiple viewpoints by first training an object class model consisting of a bank of 8 different detectors, where each detector corresponds to one of the approximate azimuth angles defined by the data set, using 41 3D CAD models. We expect our viewpoint-dependent detectors to be robust enough to cover both elevation angles. Similarly, varying distance is handled by considering part candidates at different scales. Fig. 4 visualizes 5 examples of the 8 learned models, together with corresponding example detections. It visualizes the part layouts of true positive detection hypotheses. In most cases, the hypothesized layout of object parts resembles the true layout pretty accurately, supporting exact localization at the object bounding box-level (by forming the smallest bounding box including all parts).

Comparison to state-of-the-art. We compare the performance of our model to three recent published results on the 3D Object Classes *Cars* data set, following the protocol of

[26]. Fig. 5 (a) gives precision/recall (P/R) plots for our bank of 8 detectors (green curve) and the methods of Su et al. [26] (cyan curve), Gill and Levine [11] (blue curve), and the very recent Liebelt and Schmid [14] (magenta curve). Achieving an average precision (AP) of 81.0%, our method clearly outperforms all three related approaches (APs 55.3%, 72.6%, and 76.7%). Performance can be further improved by increasing the number of detectors to 36 in a 10-degree spacing (red curve, AP 89.9%, see dense viewpoint sampling for details).

Sensitivity to viewpoint variation. In a second experiment, we examine the sensitivity of our viewpoint-based object class model to discrepancies between viewpoints used for training and testing. For this purpose, we perturb the 8 original training viewpoints systematically by $p \in \{\pm 5, \pm 10, \pm 15, \text{ and } \pm 20\}$ degrees, and test the performance of object class models consisting of all viewpoint-dependent detectors of a certain perturbation, amounting to banks of 16 detectors each (8 perturbed by $+p$ and 8 perturbed by $-p$ degrees). Fig. 5 (b) gives the corresponding P/R plots in different shades of red color, recapitulating the original green curve from Fig. 5 (a). We observe that, as expected, perturbation has a negative effect on performance in most cases, depending on the amount of perturbation. While for ±10 degrees (light red curve), performance is on par with the original bank of 8 detectors (comparing the two curves; the AP of 81.2% is in fact even slightly higher), it drops significantly for ±15 (dark orange curve, AP 70.3%) and ±20 (light orange curve, AP 58.5%) degrees. Strikingly, even for ±20 degrees, where all detectors are practically positioned as far away from the test image viewpoints as possible, the model still achieves an AP of 58.5%. The ±5 detectors (dark red curve) improve (AP 81.3%) over the original bank of 8 detectors, managing to capture slight inaccuracies in the actual test image viewpoints.

Dense viewpoint sampling. In a third experiment, we want to determine the density of sampled viewpoints (VPs) required for good performance. We thus train banks of varying numbers of detectors, each bank representing a uniform sampling of the azimuth angle range of 360 degrees into equal size intervals. Fig. 5 (c) gives P/R curves for banks of detectors with interval sizes 5, 10, 15, 20, and 30 degrees (curves in shades of red and yellow color). We start sampling the azimuth angle range at 0 degrees (corresponding to a *left* side view) for each bank, and proceed counterclockwise from there. Note that this results in different numbers of sampled VPs coinciding with test image VPs for different banks. As a consequence, the evaluation involves both viewpoint density and number of coincident VPs. In Fig. 5 (c), we observe that an interval of 30 degrees (yellow curve, 4 coincident VPs) already provides a sufficient coverage of the azimuth angle range (AP 80.7%). Performance increases consistently for denser sampling and saturates at 10 degrees (light red curve, 4 coincident VPs, AP 89.9%, outperforming related work by 13.2%). An even denser sampling of 5 degrees does not further improve performance (dark red curve, 8 coincident VPs, AP 89.8%). We observe that missing recall is often caused by missing edge information due to low image contrast (dark car color, shadows), and occurs mostly for small scale objects pictured from the most distant (*far*) VP. This holds true for 91% of the cars missed by our best performing model.

Viewpoint estimation. Fig. 5 (d) gives the confusion matrix for classifying all true positive detections according to the 8 azimuth angles defined by the data set, using the bank of 8 detectors. While we observe that neighboring VPs are rarely confused, confusion is larger for opposing views due to car symmetries (38% of *back* views are classified as *front* views). The average accuracy of 81% compares favorably to the best reported result of 70% by [14].

6 Conclusions

In this paper, we revisit the idea of learning shape models for object class recognition purely from 3D data, not using any natural training images of the object class of interest. While early approaches mostly failed in matching 3D models robustly to natural images, we benefit from intermediate advancements in object class recognition. By building our object class model on the robust combination of local part shape with a powerful model of spatial part layout, we demonstrate superior performance to state-of-the-art on a standard multi-view object class detection benchmark. While our current object class representation is based on individual per-viewpoint models, we expect integrating a continuous viewpoint estimate into a true unified 3D representation to be beneficial for performance.

Acknowledgements. This work has been funded, in part, by the DFG Emmy Noether grant GO1752/3-1.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009.

[2] M. Arie-Nachimson and R. Basri. Constructing implicit 3D shape models for pose estimation. In *ICCV*, 2009.

[3] S. Belongie, J. Malik, and J. Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *NIPS*, 2000.

[4] R.A. Brooks, R. Creiner, and T.O. Binford. The acronym model-based vision system. In *Intern. Joint Conference on Artiticial Intelligence*, pages 105–113, 1979.

[5] T. Cootes. An introduction to active shape models, 2000.

[6] Mark Everingham, Luc Gool, Christopher K. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003.

[8] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, 22(1):67–92, 1973.

[9] Y. Freund and R.E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 1997.

[10] W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov Chain Monte Carlo In Practice*. Chapman & Hall/CRC, 1996.

[11] G. Gill and M. Levine. Multi-view object detection based on spatial consistency in a low dimensional space. In *DAGM*, 2009.

[12] A. Hertzmann. Introduction to 3D non-photorealistic rendering: Silhouettes and outlines. In S. Green, editor, *SIGGRAPH 99 Course Notes*, 1999.

[13] B. Leibe, A. Leonardis, and B. Schiele. An implicit shape model for combined object categorization and segmentation. In *Toward Category-Level Object Recognition*, 2006.

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

[15] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D feature maps. In *CVPR*, 2008.

[16] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[17] D.G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.

[18] D. Marr and H.K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc. London B 200*, pages 269–194, 1978.

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

[20] A. Niculescu-Mizil and R. Caruana. Obtaining calibrated probabilities from boosting. In *UAI*, 2005.

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

[22] A.P. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28:293–331, 1986.

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

[24] S. Shalom, L. Shapira, A. Shamir, and D. Cohen-Or. Part analogies in sets of objects. In *Proceedings of Eurographics Symposium on 3D Object Retrieval*, pages 33–40, 2008.

[25] M. Stark, M. Goesele, and B. Schiele. A shape-based object class model for knowledge transfer. In *ICCV*, 2009.

[26] 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.

[27] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006.

[28] Z.W. Tu, X.R. Chen, A.L. Yuille, and S.C. Zhu. Image parsing: Unifying segmentation, detection and recognition. *ICJV*, 2005.

[29] P. Yan, S.M. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *ICCV*, 2007.

[30] S.C. Zhu, R. Zhang, and Z. Tu. Integrating bottom-up/top-down for object recognition by data driven markov chain monte carlo. In *CVPR*, 2000.