Teaching 3D Geometry to Deformable Part Models

Bojan Pepik[1] Michael Stark[1,2] Peter Gehler[3] Bernt Schiele[1]

[1]Max Planck Institute for Informatics, [2]Stanford University, [3]Max Planck Institute for Intelligent Systems

Abstract

Current object class recognition systems typically target 2D bounding box localization, encouraged by benchmark data sets, such as Pascal VOC. While this seems suitable for the detection of individual objects, higher-level applications such as 3D scene understanding or 3D object tracking would benefit from more fine-grained object hypotheses incorporating 3D geometric information, such as viewpoints or the locations of individual parts. In this paper, we help narrowing the representational gap between the ideal input of a scene understanding system and object class detector output, by designing a detector particularly tailored towards 3D geometric reasoning. In particular, we extend the successful discriminatively trained deformable part models to include both estimates of viewpoint and 3D parts that are consistent across viewpoints. We experimentally verify that adding 3D geometric information comes at minimal performance loss w.r.t. 2D bounding box localization, but outperforms prior work in 3D viewpoint estimation and ultra-wide baseline matching.

1. Introduction

Object class recognition has reached remarkable performance for a wide variety of object classes, based on the combination of robust local image features with statistical learning techniques [12, 19, 10]. Success is typically measured in terms of 2D bounding box (BB) overlap between hypothesized and ground truth objects [8] favoring algorithms implicitly or explicitly optimizing this criterion [10].

At the same time, interpretation of 3D visual scenes in their entirety is receiving increased attention. Reminiscent of the earlier days of computer vision [23, 5, 26, 22], rich, 3D geometric representations in connection with strong geometric constraints are increasingly considered a key to success [18, 7, 15, 33, 34, 2, 16]. Strikingly, there is an apparent gap between these rich 3D geometric representations and what current state-of-the-art object class detectors deliver. As a result, current scene understanding approaches are often limited to either qualitative [15] or coarse-grained quantitative geometric representations, where reasoning is typically limited to the level of entire objects [18, 15, 33, 34].

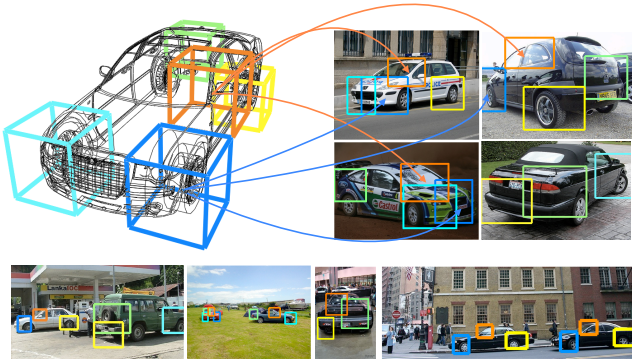

Figure 1. Example detections of our DPM-3D-Constraints. Note the correspondence of parts found across different viewpoints (color coded), achieved by a 3D parameterization of latent part positions (left). Only five parts (out of 12 parts) are shown for better readability.

The starting-point and main contribution of this paper is therefore to leave the beaten path towards 2D BB prediction, and to explicitly design an object class detector with outputs amenable to 3D geometric reasoning. By basing our implementation on the arguably most successful 2D BB-based object class detector to date, the deformable part model (DPM [10]), we ensure that the added expressiveness of our model comes at minimal loss with respect to its robust image matching to real images. To that end, we propose to successively add geometric information to our object class representation, at three different levels.

First, we rephrase the DPM as a genuine structured output prediction task, comprising estimates of both 2D object BB *and* viewpoint. This enables us to explicitly control the trade-off between accurate 2D BB localization and viewpoint estimation. Second, we enrich part and whole-object appearance models by training images rendered from CAD data. While not being as representative as real images in terms of feature statistics, these images literally come with perfect 3D annotations e.g. for position and viewpoint, which we can use to improve localization performance and viewpoint estimates.

And third, we extend the notion of discriminatively trained, deformable parts to 3D, by imposing 3D geomet-

To appear in the Proc. of the *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, June 2012.

ric constraints on the latent positions of object parts. This ensures consistency between parts across viewpoints (i.e., a part in one view corresponds to the exact same physical portion of the object in another view, see Fig. 1), and is achieved by parameterizing parts in 3D object coordinates rather than in the image plane during training. This consistency constitutes the basis of both reasoning about the spatial position of object parts in 3D and establishing part-level matches across multiple views. In contrast to prior work based on 3D shape [28, 36], our model learns 3D volumetric parts fully automatically, driven entirely by the loss function.

In an experimental study, we demonstrate two key properties of our models. First, we verify that the added expressive power w.r.t accurate object localization, viewpoint estimation and 3D object geometry does not hurt 2D detection performance too much, and even improves in some cases. In particular, we first show improved performance of our structured output prediction formulation over the original DPM for 18 of 20 classes of the challenging Pascal VOC 2007 data set [9]. We then show that our viewpoint-enabled formulation further outperforms, to the best of our knowledge, all published results on 3D Object Classes [27].

Second, we showcase the ability of our model to deliver geometrically more detailed hypotheses than just 2D BBs. Specifically, we show a performance improvement of up to 8% in viewpoint classification accuracy compared to related work on 9 classes of the 3D Object Classes data set. We then exploit the consistency between parts across viewpoints in an ultra-wide baseline matching task, where we successfully recover relative camera poses of up to 180 degrees spacing, again improving over previous work [36].

Related work. 3D geometric object class representations have been considered the holy grail of computer vision since its early days [23, 5, 26, 22], but proved difficult to match robustly to real world images. While shape-based incarnations of these representations excel in specific domains such as facial pose estimation [3] or markerless motion capture, they have been largely neglected in favor of less descriptive but robust 2D local feature-based representations for general object class recognition [12, 19, 10].

Only recently, the 3D nature of objects has again been acknowledged for multi-view object class recognition, where an angular viewpoint is to be predicted in addition to a 2D object BB. Many different methods have been suggested to efficiently capture relations between the object appearance in different viewpoints, either by feature tracking [31], image transformations [1], or probabilistic viewpoint morphing [29], and shown to deliver remarkable performance in viewpoint estimation.

While several approaches have successfully demonstrated the integration of 3D training data in the form of 3D CAD models [20, 28] and point clouds from structure-from-motion [35, 1, 13] or depth sensors [30], their outputs are typically still limited to 2D BBs and associated viewpoint estimates, conveying little information for fine-grained, geometric, scene-level reasoning. In contrast, our method outputs additional estimates of latent part positions that are guaranteed to be consistent across viewpoints, and can therefore be used to anchor part-level correspondences, thereby providing strong scene-level constraints. While this is similar in spirit to the SSfM approach of [2], we aim at fine-grained reasoning on the level of parts rather than entire objects. We note that the depth-encoded Hough voting scheme of [30] outputs depth estimates for individual features, but does not report quantitative results on how consistently these features can be found across views. On the other hand, we outperform the detailed PCA shape model of [36] in an ultra-wide baseline matching task.

On the technical side, we observe that multi-view recognition is often phrased as a two step procedure, where object localization and viewpoint estimation are performed in succession [24, 20, 13, 36]. In contrast, we formulate a coherent structured output prediction task comprising both, and simultaneously impose 3D geometric constraints on latent part positions. At the same time, and in contrast to the part-less mixture of templates model by [14], we benefit from the widely recognized discriminative power of the deformable parts framework. To our knowledge, our paper is the first to simultaneously report competitive results for 2D BB prediction, 3D viewpoint estimation, and ultra-wide baseline matching.

2. Structured learning for DPM

In the following, we briefly review the DPM model [10] and then move on to the extensions we propose in order to "teach it 3D geometry". For comparability we adopt the notation of [10] whenever appropriate.

2.1. DPM review

We are given training data $\{(I_i, y_i)\}_{1,\dots,N}$ where I denotes an image and $y = (y^l, y^b) \in \mathcal{Y}$ is a tuple of image annotations. The latter consists of y^b, the BB position of the object, *e.g.* specified through its upper, lower, left and right boundary, and $y^l \in \{-1, 1, \dots, C\}$ the class of the depicted object or -1 for background.

A DPM is a mixture of M conditional random fields (CRFs). Each component is a distribution over object hypotheses $z = (p_0, \dots, p_n)$, where the random variable $p_j = (\mathrm{u}_j, \mathrm{v}_j, l_j)$ denotes the (u, v)-position of an object part in the image plane and a level l_j of a feature pyramid image features are computed on. The root part p_0 corresponds to the BB of the object. For training examples we can identify this with y^b, whereas the parts $p_1, \dots, p_n$ are not observed and thus latent variables. We collect the two latent

variables of the model in the variable $h = \{c, p_1, \ldots, p_n\}$, where $c \in \{1, \ldots, M\}$ indexes the mixture component.

Each CRF component is star-shaped and consists of unary and pairwise potentials. The unary potentials model part appearance as HOG [6] template filters, denoted by $F_0, \ldots, F_n$. The pairwise potentials model displacement between root and part locations, using parameters (v_j, d_j), where v_j are anchor positions (fixed during training) and d_j a four-tuple defining a Gaussian displacement cost of the part p_j relative to the root location and anchor. For notational convenience we stack all parameters in a single model parameter vector for each component c, $\beta_c = (F_0, F_1, \ldots, F_n, d_1, \ldots, d_n, b)$, where b is a bias term. We denote with $\beta = (\beta_1, \ldots, \beta_M)$ the vector that contains all parameters of all mixture components. For consistent notation, the features are stacked $\Psi(I, y, h) = (\psi_1(I, y, h), \ldots, \psi_M(I, y, h))$, with $\psi_k(I, y, h) = [c = k]\psi(I, y, h)$, where $[\cdot]$ is Iverson bracket notation. The vector $\Psi(I, y, h)$ is zero except at the c'th position, so we realize $\langle \beta, \Psi(I, y, h)\rangle = \langle \beta_c, \psi(I, y, h)\rangle$. The un-normalized score of the DPM, that is the prediction function during test-time, solves $\operatorname{argmax}_{(y,h)} \langle \beta, \Psi(I, y, h)\rangle$.

2.2. Structured max-margin training (DPM-VOC)

The authors of [10] propose to learn the CRF model using the following regularized risk objective function (an instance of a latent-SVM), here written in a constrained form. Detectors for different classes are trained in a one-versus-rest way. Using the standard hinge loss, the optimization problem for class k reads

$$\begin{aligned} \min_{\beta, \xi \geq 0} \quad & \frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^{N} \xi_i \qquad (1) \\ \text{sb.t.} \quad & \forall i : y_i^l = k : \max_h \langle \beta, \Psi(I_i, y_i, h)\rangle \geq 1 - \xi_i \\ & \forall i : y_i^l \neq k : \max_h \langle \beta, \Psi(I_i, y_i, h)\rangle \leq -1 + \xi_i. \end{aligned}$$

While this has been shown to work well in practice, it is ignorant of the actual goal, 2D BB localization. In line with [4] we hence adapt a structured SVM (SSVM) formulation using margin rescaling for the loss, targeted directly towards 2D BB prediction. For a part-based model, we arrive at the following, latent-SSVM, optimization problem

$$\begin{aligned} \min_{\beta, \xi \geq 0} \quad & \frac{1}{2}\|\beta\|^2 + C \sum_{i=1}^{N} \xi_i \qquad (2) \\ \text{sb.t.} \quad & \forall i, I_i, \bar{y} \neq y_i : \max_{h_i} \langle \beta, \Psi(I_i, y_i, h_i)\rangle \\ & - \max_h \langle \beta, \Psi(I_i, \bar{y}, h)\rangle \geq \Delta(y_i, \bar{y}) - \xi_i \qquad (3) \end{aligned}$$

where $\Delta : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+$ denotes a loss function. Like in [4] we define $\Psi(I, y, h) = 0$ whenever $y^l = -1$. This has the effect to include the two constraint sets of problem (1) into this optimization program.

Based on the choice of Δ we distinguish between the following models. We use the term DPM-Hinge to refer to the DPM model as trained with objective (1) from [10] and DPM-VOC for a model trained with the loss function

$$\Delta_{\text{voc}}(y, \bar{y}) = \begin{cases} 0, & \text{if } y^l = \bar{y}^l = -1 \\ 1 - [y^l = \bar{y}^l] \frac{A(y \cap \bar{y})}{A(y \cup \bar{y})}, & \text{otherwise} \end{cases} \qquad (4)$$

first proposed in [4]. Here $A(y \cap \bar{y}), A(y \cup \bar{y})$ denote the area of intersection and union of y^b and $\bar{y}^b$.The loss is independent of the parts, as the BB is only related to the root.

Training We solve (2) using our own implementation of a gradient descent with delayed constraint generation.The latent variables render the optimization problem of the DPM a mixed integer program and we use the standard coordinate descent approach to solve it. With fixed β we find the maxima of the latent variables h_i for all training examples and also search for new violating constraints $\bar{y}, h$ in the training set. Then, for fixed latent variables and constraint set, we update β using stochastic gradient descent.

Note that the maximization step over h involves two latent variables, the mixture component c and part placements p. We search over c exhaustively by enumerating all possible values $1, \ldots, M$ and for each model solve for the best part placement using the efficient distance transform. Similar computations are needed for DPM-Hinge. Furthermore we use the same initialization for the anchor variables v and mixture components as proposed in [10] and the same hard negative mining scheme.

3. Extending the DPM towards 3D geometry

As motivated before, we aim to extend the outputs of our object class detector beyond just 2D BB. For that purpose, this section extends the DPM in two ways: a) including a viewpoint variable and b) parametrizing the entire object hypothesis in 3D. We will refer to these extensions as a) DPM-VOC+VP and b) DPM-3D-Constraints.

3.1. Introducing viewpoints (DPM-VOC+VP)

Our first extension adds a viewpoint variable to the detector output, which we seek to estimate at test time. Since several real image data sets (e.g., 3D Object Classes [27]) as well as our synthetic data come with viewpoint annotations, we assume the viewpoint observed during training, at least for a subset of the available training images. We denote with $y^v \in \{1, \ldots, K\}$ the viewpoint of an object instance, discretized into K different bins, and extend the annotation accordingly to $y = (y^l, y^b, y^v)$.

We allocate a distinct mixture component for each viewpoint, setting $c = y^v$ for all training examples carrying a

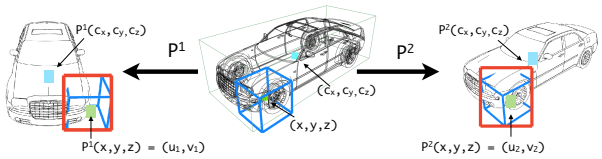

Figure 2. 3D part parametrization for an example 3D CAD model (center). Corresponding projected part positions in 2 different views, overlaid non-photorealistic renderings [28] (left, right).

viewpoint annotation. During training, we then find the optimal part placements for the single component matching the annotation for the training examples (which speeds up training). For training examples where a viewpoint is not annotated we proceed with standard DPM training, maximizing over components as well. At test time we output the estimated mixture component as a viewpoint estimate.

Since the component-viewpoint association alone does not yet encourage the model to estimate the correct viewpoint (because Eq. (3) does not penalize constraints that yield the correct BB location but a wrong viewpoint estimate), we exploit that our objective function is defined for general loss functions. We add a $0/1$ loss term for the viewpoint variables, in the following convex combination

$$\Delta_{\text{VOC+VP}}(y, \bar{y}) = (1-\alpha)\Delta_{\text{VOC}}(y, \bar{y}) + \alpha\,[y^v \neq \bar{y}^v]\,. \quad (5)$$

Note that any setting $\alpha \neq 0$ is likely to decrease 2D BB localization performance. Nevertheless we set $\alpha = 0.5$ in all experiments and show empirically that the decrease in detection performance is small, while we gain an additional viewpoint estimate. Note also, that the constraint set from Eq. (3) now include those cases where the BB location is estimated correctly but the estimated mixture component (in h) does not coincide with the true viewpoint.

A less powerful but straight-forward extension to DPM-Hinge is to use the viewpoint annotations as an initialization for the mixture components, which we refer to in our experiments as DPM-Hinge-VP.

3.2. Introducing 3D parts (DPM-3D-Constraints)

The second extension constitutes a fundamental change to the model, namely, a parametrization of latent part positions in 3D object space rather than in 2D image coordinates. It is based on the intuition that parts should really live in 3D object space rather than in the 2D image plane, and that a part is defined as a certain partial volume of a 3D object rather than as a 2D BB.

We achieve this by basing our training procedure on a set of 3D CAD models of the object class of interest that we use in addition to real training images. Being formed from triangular surface meshes, 3D CAD models provide 3D geometric descriptions of object class instances, lending themselves to 3D volumetric part parametrizations. The link to recognizing objects in 2D images is established by projecting the 3D parts to a number of distinct viewpoints, "observed" by viewpoint dependent mixture components, in analogy to DPM-VOC+VP. Since all components observe the parts through a fixed, deterministic mapping (the projections), their appearances as well as their deformations are *linked* and kept consistent across viewpoints by design.

3D Parametrization. Given a 3D CAD model of the object class of interest, we parametrize a part as an axis-aligned, 3D bounding cube of a fixed size per object class, $p_j = (\mathrm{x}_j, \mathrm{y}_j, \mathrm{z}_j)$, positioned relative to the object center (its root, see Fig. 2), in analogy to positioning parts relative to the root filter in 2D for DPM-Hinge. Further, we assume a fixed anchor position for each part p_j, from which p_j will typically move away during training, in the course of maximizing latent part positions h.

Model structure. The DPM-3D-Constraints consists of a number of viewpoint dependent mixture components, and is thus structurally equivalent to the DPM-VOC+VP. Each component observes the 3D space from a specific viewpoint c, defined by a projective mapping P^c. In full analogy to the DPM-VOC+VP, for each part p_j, each component observes i) part appearance as well as ii) part displacement. Here, both are uniquely determined by the projection $P^c(p_j)$. For i), we follow [28] to generate a non-photorealistic, gradient-based rendering of the 3D CAD model, and extract a HOG filter for the part p_j directly from that rendering. For ii), we measure the displacement between the projected root and the projected part position. Part's displacement distribution is defined in the image plane and it is independent across components. As a short-hand notation, we include the projection into the feature function $\Psi(I_i, y_i, h, P^c)$.

Learning. Switching to the 3D parametrization requires to optimize latent part placements h over object instances (possibly observed from multiple viewpoints simultaneously) rather than over individual images. Formally, we introduce an object ID variable y^o to be included in the annotation y. For a training instance y^o, we let $S(y^o) := \{i : y_i^o = y^o\}$ and compute

$$h^* = \operatorname*{argmax}_h \sum_{i \in S(y^o)} \left\langle \beta, \Psi(I_i, y_i, h, P^{y_i^v}) \right\rangle . \quad (6)$$

This problem can be solved analogously to its 2D counterpart DPM-VOC+VP: assuming a fixed placement of the object root (now in 3D), we search for the best placement h of each of the parts also in 3D. The score of the placement then depends simultaneously on all observing viewpoint-dependent components, since changing h potentially changes all projections. The computation of the maximum is still a linear operation in the number of possible 3D

AP	aero	bird	bicyc	boat	bottle	bus	car	cat	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	chair	AVG
DPM-Hinge	30.4	1.8	61.1	13.1	**30.4**	50.0	63.6	9.4	30.3	17.2	1.7	56.5	48.3	42.1	6.9	16.5	26.8	43.9	37.6	18.5	30.3
DPM-VOC	31.1	2.7	**61.3**	14.4	29.8	**51.0**	**65.7**	12.4	32.0	19.1	2.0	**58.6**	**48.8**	**42.6**	7.7	20.5	27.5	43.7	38.7	**18.7**	31.4
Vedaldi [32]	**37.6**	**15.3**	47.8	**15.3**	21.9	50.7	50.6	**30.0**	**33.0**	**22.5**	**21.5**	51.2	45.5	23.3	**12.4**	**23.9**	**28.5**	**45.3**	**48.5**	17.3	**32.1**

Table 1. 2D bounding box localization performance (in AP) on Pascal VOC 2007 [9], comparing DPM-Hinge, DPM-VOC, and [32]. Note that [32] uses a kernel combination approach that makes use of multiple complementary image features.

placements, and we use the same optimization algorithm as before: alternate between a) updating β and b) updating h and searching for violating constraints. Note that the DPM-3D-Constraints introduces additional constraints to the training examples and thereby lowers the number of free parameters of the model. We attribute performance differences to DPM-VOC+VP to this fact.

Blending with real images. Training instances for which there is only a single 2D image available (e.g., Pascal VOC data) can of course be used during training. Since there are no other examples that constrain their 3D part placements, they are treated as before in (2). Using real and synthetic images for training is called *mixed* in the experiments.

Initialization. In contrast to prior work relying on hand-labeled semantic parts [28, 36], we initialize parts in the exact data-driven fashion of the DPM, only in 3D: we choose greedily k non-overlapping parts with maximal combined appearance score (across views).

Self-occlusion reasoning. Training from CAD data allows to implement part-level self-occlusion reasoning effortlessly, using a depth buffer. In each view, we thus limit the number of parts to the l ones with largest visible area.

4. Experiments

In this section, we carefully evaluate the performance of our approach, analyzing the impact of successively adding 3D geometric information as we proceed. We first evaluate the 2D BB localization of our structured loss formulation, trained using only Δ_{VOC} (DPM-VOC, Sect. 2.2). We then add viewpoint information by optimizing for $\Delta_{\text{VOC+VP}}$ (DPM-VOC+VP, Sect. 3.1), enabling simultaneous 2D BB localization and viewpoint estimation. Next, we add synthetic training images (Sect. 3.2), improving localization and viewpoint estimation accuracy. Finally, we switch to the 3D parameterization of latent part positions during training (DPM-3D-Constraints, Sect. 3.2), and leverage the resulting consistency of parts across viewpoints in an ultra-wide baseline matching task. Where applicable, we compare to both DPM-Hinge and results of related work.

4.1. Structured learning

We commence by comparing the performance of DPM-VOC to the original DPM (DPM-Hinge), using the implementation of [11]. For this purpose, we evaluate on two diverse data sets. First, we report results for the detection task on all 20 classes of the challenging Pascal VOC 2007 data set [9]. Second, we give results on 9 classes of the 3D Object Classes data set [27], which has been proposed as a testbed for multi-view recognition, and is considered challenging because of its high variability in viewpoints (objects are imaged from 3 different distances, 3 elevations, and 8 azimuth angles). In all experiments, we use images from the respective data sets for training (sometimes in addition to our synthetic data), following the protocols established as part of the data sets [9, 27].

2D Bounding box localization. Tab. 1 gives results for 2D BB localization according to the Pascal criterion, reporting per-class average precision (AP). It compares our DPM-VOC (row 2) to the DPM-Hinge [11] (row 1) and to the more recent approach [32] (row 3), both of which are considered state-of-the-art on this data set. We first observe that DPM-VOC outperforms DPM-Hinge on 18 of 20 classes, and [32] on 8 classes. While the relative performance difference of 1.1% on average (31.4% AP vs. 30.3% AP) to DPM-Hinge is moderate in terms of numbers, it is consistent, and speaks in favor of our structured loss over the standard hinge-loss. In comparison to [32] (32.1% AP), DPM-VOC loses only 0.7% while the DPM-Hinge has 1.8% lower AP. We note that [32] exploits a variety of different features for performance, while the DPM models rely on HOG features, only.

Tab. 2 gives results for 9 3D Object Classes [27], comparing DPM-Hinge (col. 1), DPM-VOC+VP (col. 3), and DPM-Hinge-VP (col. 2), where we initialize and fix each component of the DPM-Hinge with training data from just a single viewpoint, identical to DPM-VOC+VP. We observe a clear performance ordering, improving from DPM-Hinge over DPM-Hinge-VP to DPM-VOC+VP, which wins for 5 of 9 classes. While the average improvement is not as pronounced (ranging from 88.0% over 88.4% to 88.7% AP), it confirms the benefit of structured vs. hinge-loss.

Viewpoint estimation. Tab. 2 also gives results for viewpoint estimation, phrased as a classification problem, distinguishing among 8 distinct azimuth angle classes. For DPM-Hinge, we predict the most likely viewpoint by collecting votes from training example annotations for each component. For DPM-Hinge-VP and DPM-VOC+VP, we use the (latent) viewpoint prediction. In line with previous work [27, 21], we report the mean precision in pose estimation (MPPE), equivalent to the average over the diagonal of the 8 (viewpoint) class confusion matrix. Clearly, the

AP / MPPE	DPM-Hinge	DPM-Hinge-VP	DPM-VOC+VP
iron	94.7 / 56.0	93.3 / 86.3	**96.0 / 89.7**
shoe	95.2 / 59.7	**97.9** / 71.0	96.9 / **89.8**
stapler	82.8 / 61.4	**84.4** / 62.8	83.7 / **81.2**
mouse	**77.1** / 38.6	73.1 / 62.2	72.7 / **76.3**
cellphone	60.4 / 54.6	**62.9** / 65.4	62.4 / **83.0**
head	87.6 / 46.7	89.6 / 89.3	**89.9 / 89.6**
toaster	97.4 / 45.0	96.0 / 50.0	**97.8 / 79.7**
car	99.2 / 67.1	99.6 / 92.5	**99.8 / 97.5**
bicycle	97.9 / 73.1	98.6 / 93.0	**98.8 / 97.5**
AVG	88.0 / 55.8	88.4 / 74.7	**88.7 / 87.1**

Table 2. 2D bounding box localization (in AP) and viewpoint estimation (in MPPE [21]) results on 9 3D Object classes [27].

average MPPE of 87.1% of DPM-VOC+VP outperforms DPM-Hinge-VP (74.7%) and DPM-Hinge (55.8%). It also outperforms published results of prior work [21] (79.2%) and [14] (74.0%) by a large margin of 7.9%. Initializing with per-viewpoint data already helps (59.8% vs. 74.7%), but we achieve a further boost in performance by applying a structured rather than hinge-loss (from 74.7% to 87.14%). As a side result we find that the standard DPM benefits from initializing the components to different viewpoints. We verified that fixing the components does not degrade performance, this is a stable local minima. This makes evident that different viewpoints should be modeled with different mixture components. A nice side effect is that training is faster when fixing mixture components.

Summary. We conclude that structured learning results in a modest, but consistent performance improvement for 2D BB localization. It significantly improves viewpoint estimation over DPM-Hinge as well as prior work.

4.2. Synthetic training data

In the following, we examine the impact of enriching the appearance models of parts and whole objects with synthetic training data. For that purpose, we follow [28] to generate non-photorealistic, gradient-based renderings of 3D CAD models, and compute HOG features directly on those renderings. We use 41 cars and 43 bicycle models[1] as we have CAD data from these two classes only.

2D bounding box localization. We again consider Pascal VOC 2007 and 3D Object Classes, but restrict ourselves to the two object classes most often tested by prior work on multi-view recognition [20, 28, 25, 13, 36], namely, cars and bicycles. Tab. 3 (left) gives results for Pascal cars and bicycles, comparing DPM-Hinge (col. 2) and DPM-VOC (col. 3) with the recent results of [13] (col. 1) as a reference. We compare 3 different training sets, *real*, *synthetic*, and *mixed*. First, we observe that *synthetic* performs considerably worse than *real* in all cases, which is understandable due to their apparent differences in feature statistics. Second, we observe that DPM-VOC improves significantly (from 24.7% to 34.5% AP) over DPM-Hinge for *synthetic* on cars, highlighting the importance of structured training. Third, we see that *mixed* consistently outperforms *real* for DPM-VOC, obtaining state-of-the-art performance for both cars (66.0% AP) and bicycles (61.6% AP).

Tab. 3 (right) gives results for 3D Object Classes, again training from *real*, *synthetic*, and *mixed* data, sorting results of recent prior work into the appropriate rows. In line with our findings on Pascal, we observe superior performance of DPM-VOC+VP over DPM-Hinge, as well as prior work. Surprisingly, *synthetic* (98.6% AP) performs on cars almost on par with the best reported prior result [13] (99.2%). *Mixed* improves upon their result to 99.9% AP. On bicycles, the appearance differences between *synthetic* and *real* data are more pronounced, leading to a performance drop from 98.8% to 78.1% AP, which is still superior to DPM-Hinge *synthetic* (72.2% AP) and the runner-up prior result of [20] (69.8% AP), which uses *mixed* training data.

In Fig. 3, we give a more detailed analysis of training DPM-Hinge and DPM-VOC from either *real* or *mixed* data for 3D Object Classes [27] (left) and Pascal 2007 [9] (middle, right) cars. In the precision-recall plot in Fig. 3 (middle), DPM-VOC (blue, magenta) clearly outperforms DPM-Hinge (red, green) in the high-precision region of the plot (between 0.9 and 1.0) for both *real* and *mixed*, confirming the benefit of structured max-margin training. From the recall over BB overlap at 90% precision plot in Fig. 3 (right), we further conclude that for DPM-Hinge, *mixed* (green) largely improves localization over *real* (red). For DPM-VOC, *real* (blue) is already on par with *mixed* (magenta).

Viewpoint estimation. In Tab. 3 (right), we observe different behaviors of DPM-Hinge and DPM-VOC+VP for viewpoint estimation, when considering the relative performance of *real*, *synthetic*, and *mixed*. While for DPM-VOC+VP, *real* is superior to *synthetic* for both cars and bicycles (97.5% vs. 92.9% and 97.5% vs. 86.4%), the DPM-Hinge benefits largely from synthetic training data for viewpoint classification (improving from 67.1% to 78.3% and from 73.1% to 77.5%). In this case, the difference in feature statistics can apparently be outbalanced by the more accurate viewpoints provided by *synthetic*.

For both, DPM-Hinge and DPM-VOC+VP, *mixed* beats either of *real* and *synthetic*, and switching from DPM-Hinge to DPM-VOC+VP improves performance by 11.6% for cars and 25.8% for bicycles, beating runner-up prior results by 11.8% and 18.1%, respectively.

Summary. We conclude that adding synthetic training data in fact improves the performance of both 2D BB localization and viewpoint estimation. Using *mixed* data yields state-of-the-art results for cars and bicycles on both Pascal VOC 2007 and 3D Object classes.

[1] www.doschdesign.com, www.sketchup.google.com/3dwarehouse/

		Pascal 2007 [9]			
	AP / MPPE	Glasner [13]	DPM-Hinge	DPM-VOC	DPM-3D-Const.
cars	real	32.0	63.6	65.7	-
	synthetic	-	24.7	34.5	24.9
	mixed	-	65.6	**66.0**	63.1
bicycle	real	-	61.1	61.3	-
	synthetic	-	22.6	25.2	20.7
	mixed	-	60.7	**61.6**	56.8

3D Object Classes [27]						
Liebelt [20]	Zia [36]	Payet [25]	Glasner [13]	DPM-Hinge	DPM-VOC+VP	DPM-3D-Const.
-	-	- / 86.1	99.2 / 85.3	99.2 / 67.1	99.8 / 97.5	-
-	90.4 / 84.0	-	-	92.1 / 78.3	98.6 / 92.9	94.3 / 84.9
76.7 / 70	-	-	-	99.6 / 86.3	**99.9 / 97.9**	99.7 / 96.3
-	-	- / 80.8	-	97.9 / 73.1	**98.8** / 97.5	-
-	-	-	-	72.2 / 77.5	78.1 / 86.4	72.4 / 82.0
69.8 / 75.5	-	-	-	97.3 / 73.1	97.6 / **98.9**	95.0 / 96.4

Table 3. 2D bounding box localization (in AP) on Pascal VOC 2007 [9] (left) and 3D Object Classes [27] (right). Viewpoint estimation (in MPPE [21]) on 3D Object Classes (right). Top three rows: object class car, bottom three rows: object class bicycle.

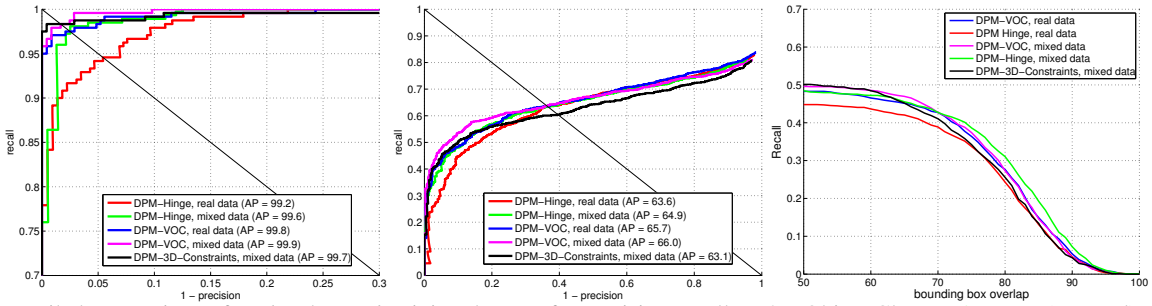

Figure 3. Detailed comparison of *real* and *mixed* training data. Left: Precision-recall on 3D Object Classes [27] cars (zoomed). Middle: Precision-recall on Pascal VOC 2007 [9] cars. Right: Recall over bounding box overlap at 90% precision on Pascal 2007 cars.

4.3. 3D deformable parts

We finally present results for the DPM-3D-Constraints, constraining latent part positions to be consistent across viewpoints. We first verify that this added geometric expressiveness comes at little cost w.r.t. 2D BB localization and viewpoint estimation, and then move on to the more challenging task of ultra-wide baseline matching, which is only enabled by enforcing across-viewpoint constraints.

2D bounding box localization. In Tab. 3 (left, last col.), we observe a noticeable performance drop from DPM-VOC to DPM-3D-Constraints for both Pascal cars and bicycles for *synthetic* (from 34.5% to 24.9% and 25.2% to 20.7% AP, respectively). Interestingly, this drop is almost entirely compensated by *mixed*, leaving us with remarkable 63.1% AP for cars and 56.8% AP for bicycles, close to the state-of-the-art results (DPM-Hinge). Tab. 3 (right, last col.) confirms this result for 3D Object Classes. DPM-3D-Constraints obtains 0.2% lower AP for cars and 2.6% lower AP for bicycles, maintaining performance on par with the state-of-the-art.

Viewpoint estimation. The switch from DPM-VOC+VP to DPM-3D-Constraints results in performance drop, which we attribute to the reduced number of parameters due to the additional 3D constraints. Still, this performance drop is rather small. In particular for *mixed* (we lose only 1.3% MPPE for cars and 2.5% for bicycles).

Ultra-wide baseline matching. In this experiment, we quantify the ability of the DPM-3D-Constraints to hypothesize part positions that are consistent across viewpoints. We adapt the experimental setting proposed by [36], and use corresponding part positions on two views of the same object as inputs to a structure-from-motion (SfM) algorithm. We then measure the Sampson error [17] of the resulting fundamental matrix(see Fig. 4), using ground truth correspondences. We use the same subset of 3D Object Classes cars as [36], yielding 134 image pairs, each depicting the same object from different views, against static background. Tab. 4 gives the results (percentage of estimated fundamental matrices with a Sampson error < 20 pixels), comparing a simple baseline using SIFT point matches (col. 1) to the results by [36] (col. 2), and the DPM-3D-Constraints using 12 (col. 3) and 20 parts (col. 4), respectively, for varying angular baselines between views. As expected, the SIFT baseline fails for views with larger baselines than 45°, since the appearance of point features changes too much to provide matches. On the other hand, we observe competitive performance of our 20 part DPM-3D-Constraints compared to [36] for baselines between 45° and 135°, and a significant improvement of 29.4% for the widest baseline (180°), which we attribute to the ability of our DPM-3D-Constraints to robustly distinguish between opposite view points, while [36] reports confusion for those cases.

Azimuth	SIFT	Zia [36]	DPM-3D-Const. 12	DPM-3D-Const. 20
45 °	2.0%	55.0%	49.1%	54.7%
90 °	0.0%	60.0%	42.9%	51.4%
135 °	0.0%	52.0%	55.2%	51.7%
180 °	0.0%	41.0%	52.9%	70.6%
AVG	0.5%	52.0%	50.0%	57.1%

Table 4. Ultra-wide baseline matching performance, measured as fraction of correctly estimated fundamental matrices. Results for DPM-3D-Const. with 12 and 20 parts versus state-of-the-art.

Figure 4. Example ultra-wide baseline matching [36] output. Estimated epipoles and epipolar lines (colors correspond) for image pairs.

Summary. Our results confirm that the DPM-3D-Constraints provides robust estimates of part positions that are consistent across viewpoints, and hence lend themselves to 3D geometric reasoning. At the same time, the DPM-3D-Constraints maintains performance on par with state-of-the-art for both 2D BB localization and viewpoint estimation.

5. Conclusions

We have shown how to teach 3D geometry to DPMs, aiming to narrow the representational gap between state-of-the-art object class detection and scene-level, 3D geometric reasoning. By adding geometric information on three different levels, we improved performance over the original DPM and prior work. We achieved improvements for 2D bounding box localization, viewpoint estimation, and ultra-wide baseline matching, confirming the ability of our model to deliver more expressive hypotheses w.r.t. 3D geometry than prior work, while maintaining or even increasing state-of-the-art performance in 2D bounding box localization .

Acknowledgements This work has been supported by the Max Planck Center for Visual Computing and Communication. We thank M. Zeeshan Zia for his help in conducting wide baseline matching experiments.

References

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

[2] S. Y. Bao and S. Savarese. Semantic structure from motion. In *CVPR*, 2011. 1, 2

[3] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *PAMI*, 2003. 2

[4] M. Blaschko and C. Lampert. Learning to localize objects with structured output regression. In *ECCV*, 2008. 3

[5] R. Brooks. Symbolic reasoning among 3-d models and 2-d images. *Artificial Intelligence*, 17:285–348, 1981. 1, 2

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

[7] A. Ess, B. Leibe, K. Schindler, and L. V. Gool. Robust multi-person tracking from a mobile platform. *PAMI*, 2009. 1

[8] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL VOC 2007 Results. 2, 5, 6, 7

[10] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2009. 1, 2, 3

[11] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://www.cs.brown.edu/ pff/latent-release4/. 5

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, 2003. 1, 2

[13] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, 2011. 2, 6, 7

[14] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV*, 2010. 2, 6

[15] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 1

[16] A. Gupta, S. Satkin, A. A. Efros, and M. Hebert. From 3D scene geometry to human workspace. In *CVPR*, 2011. 1

[17] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. 7

[18] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 2008. 1

[19] B. Leibe, A. Leonardis, and B. Schiele. An implicit shape model for combined object categorization and segmentation. In *Toward Category-Level Object Recognition*, 2006. 1, 2

[20] J. Liebelt and C. Schmid. Multi-view object class detection with a 3D geometric model. In *CVPR*, 2010. 2, 6, 7

[21] R. J. Lopez-Sastre, T. Tuytelaars, and S. Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *ICCV-WS CORP*, 2011. 5, 6, 7

[22] D. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 1987. 1, 2

[23] D. Marr and H. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *Proc. Roy. Soc. London B*, 200(1140):269–194, 1978. 1, 2

[24] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR*, 2009. 2

[25] N. Payet and S. Todorovic. From contours to 3d object detection and pose estimation. In *ICCV*, 2011. 6, 7

[26] A. Pentland. Perceptual organization and the representation of natural form. *Artificial Intelligence*, 28, 1986. 1, 2

[27] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV*, 2007. 2, 3, 5, 6, 7

[28] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, 2010. 2, 4, 5, 6

[29] 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. 2

[30] M. Sun, B. Xu, G. Bradski, and S. Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV'10*. 2

[31] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR*, 2006. 2

[32] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 5

[33] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV'10*. 1

[34] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *ECCV*, 2010. 1

[35] P. Yan, S. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *ICCV*, 2007. 2

[36] Z. Zia, M. Stark, K. Schindler, and B. Schiele. Revisiting 3d geometric models for accurate object shape and pose. In *3dRR-11*, 2011. 2, 5, 6, 7, 8