# Multi-View and 3D Deformable Part Models

Bojan Pepik, *Student Member, IEEE*, Michael Stark, *Member, IEEE*, Peter Gehler, *Member, IEEE*, and Bernt Schiele, *Member, IEEE*

**Abstract**—As objects are inherently 3D, they have been modeled in 3D in the early days of computer vision. Due to the ambiguities arising from mapping 2D features to 3D models, 3D object representations have been neglected and 2D feature-based models are the predominant paradigm in object detection nowadays. While such models have achieved outstanding bounding box detection performance, they come with limited expressiveness, as they are clearly limited in their capability of reasoning about 3D shape or viewpoints. In this work, we bring the worlds of 3D and 2D object representations closer, by building an object detector which leverages the expressive power of 3D object representations while at the same time can be robustly matched to image evidence. To that end, we gradually extend the successful deformable part model [1] to include viewpoint information and part-level 3D geometry information, resulting in several different models with different level of expressiveness. We end up with a 3D object model, consisting of multiple object parts represented in 3D and a continuous appearance model. We experimentally verify that our models, while providing richer object hypotheses than the 2D object models, provide consistently better joint object localization and viewpoint estimation than the state-of-the-art multi-view and 3D object detectors on various benchmarks (KITTI [2], 3D object classes [3], Pascal3D+ [4], Pascal VOC 2007 [5], EPFL multi-view cars [6]).

**Index Terms**—Object detection, 3D object models, deformable part models, structured output learning

## 1 Introduction

Object class detection has reached remarkable performance for a wide variety of object classes, based on the combination of robust local image features with statistical learning techniques [1], [7], [8], [9], [10]. Success is typically measured in terms of 2D bounding box (BB) overlap between hypothesized and ground truth objects [11] favoring algorithms implicitly or explicitly optimizing this criterion [1].

Although the state-of-the-art methods for object class detection are appearance based, in the early days of computer vision, geometry based 3D representations of objects and entire scenes were considered the holy grail [12], [13], [14], [15]. Being more compact and providing a more faithful approximation of the physical world than 2D image projections, they were deemed more powerful w.r.t. reasoning about individual objects, their interactions in complete scenes, and even functions [16], [17]. Despite being rich, these representations could not be reliably matched to real-world imagery. As a consequence, they were largely neglected in favor of 2D appearance based representations of object classes. Recently, researchers have reconsidered the 3D nature of the vision problem in the context of scene understanding. Here, 3D information has shown to be valuable to reduce false detections [18], [19], [20]. This has also fueled the development of multi-view recognition methods [3], [21], [22], [23], [24], [25], [26], [27], [28], [29], providing richer object hypotheses in the form of viewpoint estimates as additional cue for scene-level reasoning [30], [31], [32]. However, most approaches are still either limited with respect to the degree of 3D modeling, or can not provide competitive performance in terms of 2D BB localization. In particular, the ability to provide richer object hypotheses than 2D BB is typically associated with sacrificing 2D localization performance in comparison to state-of-the-art object detectors.

In this work, we aim to combine the best of both worlds, namely, to leverage performance from one of the most powerful appearance based 2D object class detectors to date, and a geometry based 3D object class representation that allows for fine-grained 3D object and scene reasoning. In this way, we hope to benefit from the natural, compact and rich 3D representation while retaining the robustness in matching to real-world images. The goal is 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 one of the arguably most successful 2D BB-based object class detectors to date, the deformable part model (DPM) [1], we ensure that the added expressiveness of our model comes at minimal loss with respect to its robust matching to real images. To that end, we propose to successively add geometric information to our object class representation, at four 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 introduce 3D geometric constraints on the latent positions of object parts in the DPM. This ensures consistency between parts across viewpoints

- *B. Pepik, M. Stark, and B. Schiele are with the Max Planck Institute for Informatics, 66123 Saarbrücken, Germany. E-mail: {bpepikj, stark, schiele}@mpi-inf.mpg.de.*
- *P. Gehler is with the Max Planck Institute for Intelligent Systems, 72076 Tübingen, Germany. E-mail: pgehler@tuebingen.mpg.de.*

*Manuscript received 18 July 2014; revised 5 Feb. 2015; accepted 22 Feb. 2015. Date of publication 2 Mar. 2015; date of current version 7 Oct. 2015.*
*Recommended for acceptance by D. Ramanan.*
*For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.*
*Digital Object Identifier no. 10.1109/TPAMI.2015.2408347*

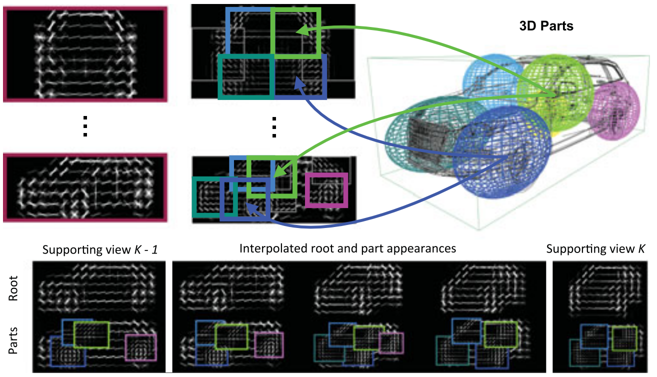


Fig. 1. 3D$^2$PM model visualization. Learned part 3D displacement distributions along with the continuous appearance model.

(i.e., a part in one view corresponds to the exact same physical portion of the object in another view). Third, we extend the notion of discriminatively trained, deformable parts to 3D, by explicitly parametrizing the parts positions and displacement distributions in 3D object coordinates rather than in the image plane (see Fig. 1). And fourth, we introduce a continuous appearance model (see Fig. 1), which allows for arbitrarily fine viewpoint estimates in contrast to state-of-the-art multi-view detection methods which can predict only a discrete set of viewpoint classes.

In this work we make the following specific contributions. First, we propose a 3D extension of the powerful DPM, combining the representational power of 3D modeling with robust matching to real-world images. Second, we demonstrate that our models deliver richer object hypotheses than 2D BB, in the form of viewpoint estimates of arbitrary granularity and part localization consistent across viewpoints, outperforming prior work various datasets. Third, in contrast to previous work on 3D object models, we show competitive performance to state-of-the-art techniques for 2D BB localization. Fourth, we use 3D CAD data of the object class of interest mainly as a 3D geometry cue, as well as to enrich the appearance model with rendered images from CAD data. While being not as representative as real world images in terms of feature statistics, these images come with perfect BB and viewpoint annotation, which we can use to improve localization performance and viewpoint estimates.

## 2 Related Work

Object class detection is at the core of many computer vision problems, and as such has been addressed since the beginnings of computer vision. 2D features-based representations for general object class recognition [1], [9], [10], [33], [34], [35], [36], have been the dominant paradigm in object detection in recent years. Multi-view detectors and 3D object representations have been receiving increasing attention recently, due to its potential to aid scene-level reasoning [30], [31], [32]. In the following we review these groups of object detection methods.

### 2.1 Bounding Box Object Detection

Inspired by the challenging object detection benchmarks (Pascal VOC [11], ImageNet [37], SUN [38]) BB driven object class detectors have achieved fascinating detection performance. Combining precomputed (SIFT [39], HOG [40]) or directly learned from data (CNN [41], OverFeat [10]) image features, with discriminative learning techniques (SVM [42], AdaBoost [43], Random Forests [44]) these methods have been the main driving force in object detection. Methods like the implicit shape model [8], which spatially aggregates object votes, or the DPM [1], representing the object as a collection of 2D parts have emerged. While these methods are using precomputed image features, recently, due to the arrival of large datasets (ImageNet), deep learning based methods that directly learn features from raw images (RCNN [9], OverFeat [10], DetectorNet [45]) have become popular.

While BB oriented detectors are the de-facto state-of-the-art in object detection, they provide very limited output, consisting of a BB and a class label, completely ignoring the 3D nature of objects. On the other hand, models with higher expressiveness have the potential to boost higher-level tasks like 3D scene understanding. In this work we want to overcome the limitations of these methods. To that end, we aim to construct an object detector that retains the matching ability of the BB oriented detectors while being aware of the 3D geometry of the object class of interest (see Section 3.5).

### 2.2 Multi-View Object Detection

The interest in richer object hypotheses has inspired the creation of several multi-view detection benchmarks (3D object classes [3], EPFL multi-view cars [6], Pascal3D+ [4], KITTI [2]). The best performing multi-view detectors usually model object classes as a collection of distinct views, forming a bank of viewpoint-dependent detectors. The form of these detectors is typically inspired by existing approaches from the literature that have proven to perform well for the single view case. Variants include shape templates [27], implicit shape models [21], HOG templates [6], constellation models [25] and DPM [4], [29], [46], [47], [48], [49]. Neighboring views are either treated independently [6], [25], connected by means of feature tracking [21], or considered jointly in a convex optimization framework [27], [29], [46], [50].

While multi-view approaches achieve remarkable results in predicting a discrete set of object poses [4], [29], [46], they have several limitations. First, they usually treat the discrete views independently [25], [29], [46]. Second, they typically require evaluating a large number of view-based detectors , resulting in considerable runtime complexity (e.g., 32 shape templates [27], 36 constellation models [25]). In contrast, our 3D object detectors (see Sections 3.4 and 3.5) establishes part correspondences across views and are able to synthesize appearance models for viewpoints of arbitrary granularity on the fly. This results to significant speed-ups (see Section 4.4).

### 2.3 3D Object Class Representations

Other methods acknowledge the 3D nature of objects and maintain an explicit representation of the 3D placement of individual features [3], [22], [28], [51], [52], or object parts [23], [24], [26], [47], [48], [53], [54]. 3D geometry is either provided in the form of a depth sensor [22], [51], [52], structure-from-motion [28], [48], [53], [55], or 3D CAD models [24], [26], [54], [56] during training, and modeled either non-parametrically [22], [28], [51], [52], in the form of 3D Gaussians [24], or 3D wireframe models [26], [53].

While these 3D representations constitute more compact and more faithful descriptions of object classes than their 2D counterparts, they typically can not compete with modern object class detectors optimized for 2D BB localization such as the DPM or RCNN. In our work we aim to overcome this limitation of 3D object class representations by designing a 3D extension of the DPM. To that end, we gradually reformulate the model as a 3D DPM by representing part appearance as well as positions in 3D. As a consequence, our formulation models part positions as true 3D distributions and allows to synthesize part appearance models for arbitrary viewpoints.

### 2.4 Model Learning

From the perspective of learning, the BB oriented object detectors (DPM, RCNN) are typically trained using loss functions that are tuned for classification, sometimes even using a pre-selected set of BBs to detect on (e.g., RCNN uses proposal regions from [36]). In addition, multi-view recognition is often phrased as a multi-class classification problem [4], [27], [29] ignoring the continuous nature of the viewpoint variable.

In contrast, we formulate a coherent structured output learning framework comprising both objectives: (i) 2D object localization and (ii) viewpoint estimation. The joint consideration of the two tasks at hand leads to consistently better object localization and viewpoint estimation on several datasets (see Section 4.2).

## 3 Multi-View and 3D Deformable Part Models

In this section we introduce our geometry-aware multi-view and 3D object models. We start with the well-known DPM [1] and gradually introduce 3D geometry cues. This results in a 3D object model, a full 3D extension of DPM. The resulting model parameterizes part positions and distributions in 3D and has a continuous appearance model. We refer to it as 3D$^2$PM. Because we encode the underlying 3D object structure, the model becomes more compact with a smaller total number of parameters compared to the DPM. At the same time, we obtain a model that is more descriptive of the 3D object of interest.

We describe our models successively. First, in Section 3.1 we introduce notation and the idea behind a part-based model. After revisiting the DPM of [1] in Section 3.2, we introduce the 2D DPM-VOC+VP in Section 3.3, a multi-view object detector which in contrast to the DPM predicts object viewpoint, in addition to the 2D BB. We proceed by introducing 3D geometry into the model and in Section 3.4 present DPM-3D-Constraints that leverages 3D part constraints, by parameterizing part positions in 3D object coordinates. This establishes part correspondences across different views of the same object. In Section 3.5 we introduce the 3D object model 3D$^2$PM, parameterizing part positions, as well as displacement distributions in 3D. The 3D$^2$PM includes a continuous appearance model.

### 3.1 Part Based Models Preliminaries

We are given data $\{X\}_{1,\dots,N}$ where $X$ represents an object, defined in image space, or in 3D object coordinates, like a CAD model. The idea behind part-based models is to represent an object by a collection of parts [1], [57], [58]. Previous work has considered different spatial configurations of parts, ranging from star-shaped [1], tree-shaped [58], to fully-connected constellations [25]. Here we build upon the view of a generalized deformable part model as a star-shaped conditional random field (CRF). A star-shaped CRF defines a distribution over object and part positions $\mathbf{o} = (o_0, \dots, o_P)^1$ where $o_i$ denotes an object part, with $o_0$ being the whole object or the root node and the rest being the child nodes. We define an object part as an axis-aligned hypercube. An object part can be defined in 3D object coordinates as $p_i = [x_1, y_1, z_1, x_2, y_2, z_2]$ in which case the CRF defines a distribution over 3D bounding cubes, or in the image plane as $q_i = [u_1, v_1, u_2, v_2]$, defining a distribution over 2D BBs. Thus, given an object $X$ and a star-shaped CRF model $\boldsymbol{\theta}$, the joint probability distribution over the object hypotheses reads

$$p(\mathbf{o}|\boldsymbol{\theta}, X) \propto \prod_{i=0}^{P} \Psi^u(o_i, \alpha_i, X) \prod_{i=1}^{P} \Psi^p(o_i, o_0, \beta_i). \tag{1}$$

This distribution decouples in part-wise terms and for each part $o_i$ there are two terms. First, the unary factor $\Psi^u(o_i, \alpha_i, X)$ scores an object part hypothesis $o_i$, given the object $X$. This unary factor is also referred to as the "appearance" term as it captures the appearance of an object part. The second factor is a pairwise term, $\Psi^p(o_i, o_0, \beta_i)$, referred to as the "spatial" term. The pairwise term specifies part $o_i$ placements w.r.t. the root part $o_o$. All factors are log-linear. We denote the full set of parameters by $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$ that includes parameters of the unary $\boldsymbol{\alpha} = [\alpha_0, \dots, \alpha_P]$ and pairwise terms $\boldsymbol{\beta} = [\beta_1, \dots, \beta_P]$. For the feature functions we write $\boldsymbol{\phi} = [\phi(o_0, X), \dots, \phi(o_P, X)]$ for the unaries and $\boldsymbol{\eta} = [\eta(o_1, o_0), \dots, \eta(o_p, o_0)]$ for the pairwise features respectively, so the energy of the CRF in this general form reads

$$\langle \boldsymbol{\theta}, \boldsymbol{\psi} \rangle = \langle \boldsymbol{\alpha}, \boldsymbol{\phi} \rangle + \langle \boldsymbol{\beta}, \boldsymbol{\eta} \rangle. \tag{2}$$

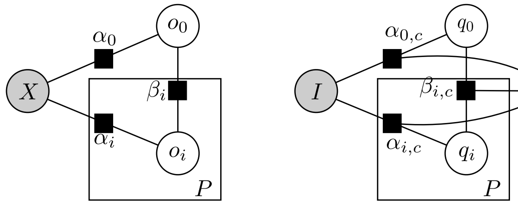
In the following sections we specify the unary and pairwise terms for each of the models. Fig. 2a depicts the graphical model defined by the star-shaped CRF.

Previous work on object detection with part-based models (e.g., the DPM [1]) defines a distribution over 2D object hypotheses by parameterizing parts, unary and pairwise terms in the 2D image space. The models that we present in this work extend the DPM, and gradually shift the parameterization from 2D image space to 3D object space, resulting in an object model parameterized entirely in 3D.

### 3.2 DPM-Hinge

The 2D part-based model of [1] is one of the most successful object detectors nowadays, as evidenced by its performance on benchmark datasets [59] and its use as a building block

[1] We use regular font characters to denote part parameters of features. We use characters with bold font whenever we stack parameters from multiple parts or components.



(a) star-shaped CRF (b) DPM [1]

Fig. 2. Graphical models depicting (a) general part-based model as a CRF over the parts $o_i$ conditioned on the data $X$. (b) The 2D DPM, conditioned on an image $I$. With shaded nodes, we denote the observed variables.

in many subsequent works. Given an image, the DPM outputs a set of 2D BBs, coarsely localizing the objects. In the remainder of this paper we will refer to the DPM version of [1] as DPM-Hinge, as it uses the hinge loss during model learning and allows us to distinguish it from the other models.

*Representation.* DPM-Hinge is a mixture model with $C$ components, defined in 2D image space. Each component $c \in \{1, \dots, C\}$ captures the appearance and part placement of an object in a particular aspect (often coinciding with viewpoint). DPM-Hinge parameterizes an object hypothesis as a collection of 2D BBs of the object $q_0$ and it's parts $q_1, \dots, q_P$. For an image $I$, the score of component $c$ for an object hypothesis $\mathbf{q} = [q_0, \dots, q_P]$ is defined as

$$\langle \boldsymbol{\theta}_c, \boldsymbol{\psi}_c(\mathbf{q}, I) \rangle = \sum_{i=0}^{P} \langle \alpha_{i,c}, \phi(q_i, I) \rangle + \sum_{i=1}^{P} \langle \beta_{i,c}, \eta(q_i, q_0) \rangle, \tag{3}$$

where $\boldsymbol{\theta}_c = [\boldsymbol{\alpha}_c, \boldsymbol{\beta}_c]$ denote unary $\boldsymbol{\alpha}_c$ and pairwise $\boldsymbol{\beta}_c$ parameters of component $c$. In particular, the parameters collect per-part $q_i$ variables as $\boldsymbol{\alpha}_c = [\alpha_{0,c}, \dots, \alpha_{P,c}]$ and $\boldsymbol{\beta}_c = [\beta_{1,c}, \dots, \beta_{P,c}]$. The unary part parameters $\alpha_{i,c}$ are 2D filters for the HOG [40] appearance features $\phi(q_i, I)$. The pairwise factor corresponds to a Gaussian distribution over the part $q_i$ placement relative to $q_0$. The feature function computes the natural parameters of a 2D Gaussian $\mathcal{N}(q_i \mid q_0, \mu_{i,c}, \Sigma_{i,c})$. The pairwise features are defined as $\eta_i(q_i, q_0) = -[du_i, dv_i, du_i^2, dv_i^2]$, where $[du_i, dv_i] = q_i - (2q_0 + j_i)$[2]. Here, $j_i$ represents the anchor part position relative to the root. The variables can be understood as $\beta_{i,c} = [\mu_{i,c}^u, \mu_{i,c}^v, \sigma_{i,c}^u, \sigma_{i,c}^v]$, the parameters of a 2D Gaussian.

For the full DPM-Hinge model all parameters from all mixture components are stacked $\boldsymbol{\theta} = [\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_C]$. The graphical model depicting the DPM-Hinge is illustrated in Fig. 2b.

*Inference.* During inference [1] computes the maximum-a-posteriori (MAP) estimate over object hypotheses and components $c^*, \mathbf{q}^* = \operatorname{argmax}_{c,\mathbf{q}} \langle \boldsymbol{\theta}_c, \boldsymbol{\psi}_c(\mathbf{q}, I) \rangle$. This problem involves maximization over two variables, the discrete mixture component $c$ and all part placements $\mathbf{q}$. For each component $c$ the part placement can be found using the efficient distance transform, and the search over $c$ is done by exhaustive enumeration [1].

*Learning.* For parameter estimation, the training data is available in pairs $\{(I_i, y_i)\}_{i=1,\dots,N}$ where $I$ is an image and $y = (y^l, y^b) \in \mathcal{Y}$ is a tuple of annotations. The annotation includes an object class label $y^l \in \{-1, 1, \dots, L\}$, and a 2D BB $y^b$.

Felzenszwalb et al. [1] propose to learn the free parameters of their model using a regularized risk objective with the hinge loss. For every object class $k \in \{1, \dots, L\}$ there is a separate optimization problem

$$\begin{aligned} \min_{\boldsymbol{\theta}, \xi \geq 0} \quad & \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{sb.t.} \quad & \forall i : y_i^l = k : \max_{c, \mathbf{q}_{y_i}} \langle \boldsymbol{\theta}_c, \boldsymbol{\psi}_c(\mathbf{q}_{y_i}, I_i) \rangle \geq 1 - \xi_i \\ & \forall i : y_i^l \neq k : \max_{c, \mathbf{q}_{y_i}} \langle \boldsymbol{\theta}_c, \boldsymbol{\psi}_c(\mathbf{q}_{y_i}, I_i) \rangle \leq -1 + \xi_i, \end{aligned} \tag{4}$$

where $\mathbf{q}_{y_i} = [y_i^b, q_1 \dots, q_P]$, where $y^b$ is the BB of the example and fixed for every training example. The part positions $q_i$ are latent variables, because part annotations are not available. In [1] initial values for the component assignments are obtained via aspect ratio clustering and are kept latent during training. This problem is a latent SVM [1] with hinge loss, which is the reason we refer to the DPM as DPM-Hinge.

### 3.3 DPM-VOC+VP

The DPM-Hinge has shown remarkable performance in terms of 2D object localization, it is however not designed to predict the viewpoint of an object. A multi-view object detector could boost object detection quality and it could be beneficial for high level tasks like 3D scene understanding [31]. The first extension we introduce, DPM-VOC+VP, augments DPM-Hinge output with a viewpoint variable $v$.

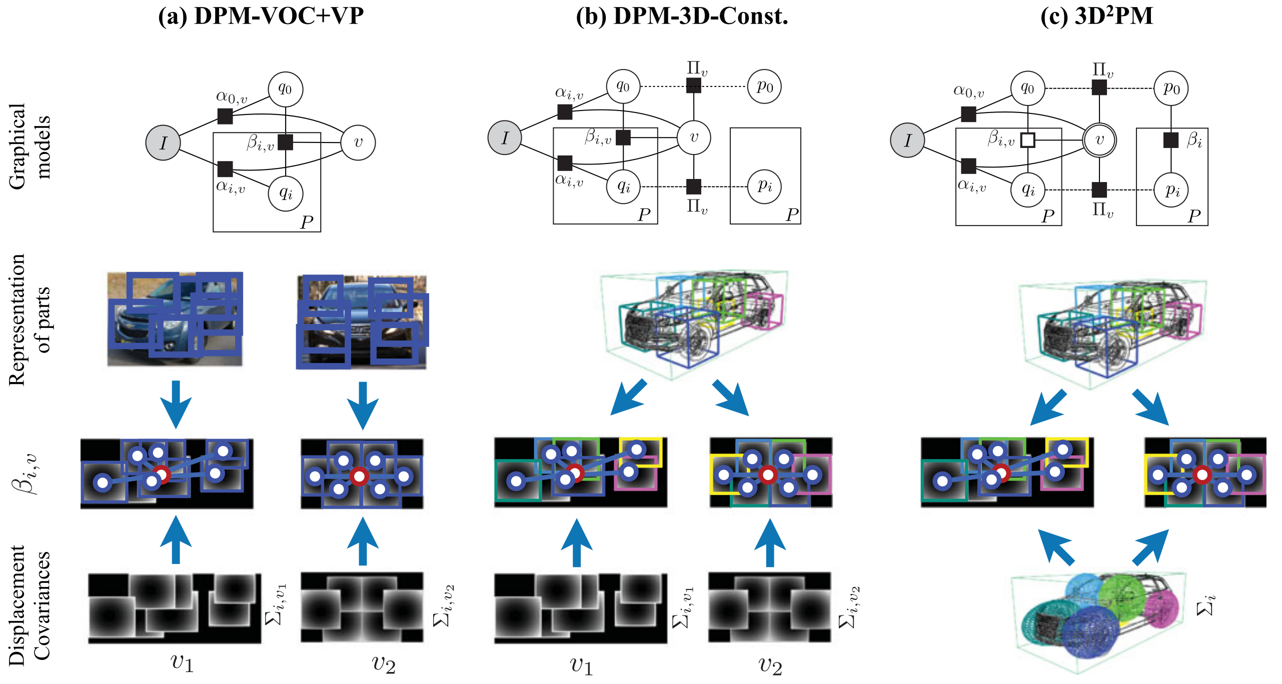
*Representation.* In DPM-VOC+VP we allocate a separate mixture component to each discrete viewpoint $v$. Every viewpoint component $\boldsymbol{\theta}_v = [\boldsymbol{\alpha}_v, \boldsymbol{\beta}_v]$, has its own unary $\boldsymbol{\alpha}_v = [\alpha_{0,v}, \dots, \alpha_{P,v}]$ and pairwise $\boldsymbol{\beta}_v = [\beta_{0,v}, \dots, \beta_{P,v}]$ parameters. DPM-VOC+VP has the same CRF structure as the DPM-Hinge. In addition, it explicitly encodes the object viewpoint $v$. Fig. 3a illustrates the DPM-VOC+VP model.

*Inference.* The inference is the same as for DPM-Hinge, a MAP estimate over viewpoints and BBs $\mathbf{q}^*, v^* = \operatorname{argmax}_{v,\mathbf{q}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}_v(\mathbf{q}, I) \rangle$. We use the same inference technique as DPM-Hinge.

*Learning.* Since we are interested in joint object 2D localization and viewpoint estimation, we leverage viewpoint annotations in the datasets. We denote the viewpoint class label of a given training example as $y^v \in \{1, \dots, K\}$, in addition to the BB $y^b$ and the class $y^l$ labels. In contrast to the DPM-Hinge, there is a semantic meaning to the selected component and thus it must be chosen correctly. Therefore we adapt a structured SVM [60] with margin rescaling for optimization. This objective has previously been proposed for BB detection in [61]. The final latent-SSVM optimization problem is

$$\begin{aligned} \min_{\boldsymbol{\theta}, \xi \geq 0} \quad & \frac{1}{2}\|\boldsymbol{\theta}\|^2 + C \sum_{i=1}^{N} \xi_i \\ \text{sb.t.} \quad & \forall i, \bar{y} \neq y_i : \max_{\mathbf{q}_{y_i}} \langle \boldsymbol{\theta}_{y_i^v}, \boldsymbol{\psi}_{y_i^v}(\mathbf{q}_{y_i}, I_i) \rangle \\ & \qquad - \max_{\bar{v}, \bar{\mathbf{q}}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}_{\bar{v}}(\bar{\mathbf{q}}, I_i) \rangle \geq \Delta(y_i, \bar{y}) - \xi_i, \end{aligned} \tag{5}$$

[2] We use the upper left corners of the parts to compute the displacement features.


(a) DPM-VOC+VP

(b) DPM-3D-Const.

(c) 3D²PM

Fig. 3. Comparison of the different presented models. In the first row from left to right the graphical models of (a) DPM-VOC+VP, (b) DPM-3D-Constraints, and (c) 3D²PM, are shown. In the second row, the part parameterization is illustrated. The third row shows a possible layout of the part configuration. The last row visualizes the covariances of the placement distributions. The variables $\beta_{i,v}$, of the 3D²PM are implicitly defined via projection, see Section 3.5. Both DPM-3D-Constraints and 3D²PM define parts in a 3D reference frame, therefore it is possible to establish part-correspondences across different viewpoints.

where $\mathbf{q}_{y_i} = [y_i^b, q_1, \ldots, q_P]$ as before is the annotated object BB $y_i^b$ with the latent part placements and $\bar{\mathbf{q}}_i = [\bar{y}^b, \bar{q}_1, \ldots, \bar{q}_P]$ a different object hypothesis. Note that for the positive training examples, the viewpoint component is observed. Like in [61] we define $\boldsymbol{\psi}_v(\mathbf{q}_{y_i}, I_i) = 0$ whenever $y_i^l = -1$. This has the effect to include the two constraint sets of problem (4) into this optimization problem.

The loss function $\Delta$ is defined on both the predicted BBs and viewpoint at the same time. We use a convex combination of a BB localization $\Delta_{VOC}$ and viewpoint loss $\Delta_{VP}$, namely $\Delta(y, \bar{y}) = \gamma\Delta_{VOC}(y, \bar{y}) + (1-\gamma)\Delta_{VP}(y, \bar{y})$, with $\gamma \in [0, 1]$.

The performance measure for BB accuracy in the standard benchmarks is the intersection over union score $A(y \cap \bar{y})/A(y \cup \bar{y})$ of two BBs $y, \bar{y}$. Therefore, as proposed in [61] we use the following loss function as a proxy:

$$\Delta_{\text{VOC}}(y, \bar{y}) = \begin{cases} 0, & \text{if } y^l = \bar{y}^l = -1 \\ 1 - [y^l = \bar{y}^l]\frac{A(y \cap \bar{y})}{A(y \cup \bar{y})}, & \text{otherwise.} \end{cases} \tag{6}$$

The viewpoint loss $\Delta_{VP}$ is the $0/1$ classification error with different discrete viewpoint predictions treated as different classes.

In case only the location of the object is of interest, one can set $\gamma = 1$, in which case we refer to the model as DPM-VOC, which uses the same initialization, based on aspect ratio clustering, as for the DPM-Hinge. If both tasks are of interest, we set $\gamma = 0.5$ and refer to the resulting model as DPM-VOC+VP.

**Algorithm 1.** DPM-VOC+VP training algorithm

```
Input: $\{I_i, y_i\}_1^N$ $I_i$ is an image, $y_i$ annotations
Output: Trained DPM-VOC+VP $\boldsymbol{\theta}$
$\boldsymbol{\theta} \leftarrow$ InitModel ($pos, neg$)
$\mathcal{P} = \emptyset,\ \mathcal{S} = \emptyset,\ \mathcal{N} = \emptyset$
while outer loop do
  //Find optimal parts for each positive example
  foreach $i \in pos$ do
    $\mathbf{q}_i \leftarrow \text{argmax}_{\mathbf{q}_i} \langle \boldsymbol{\theta}_{v_i}, \boldsymbol{\psi}_{v_i}(\mathbf{q}_i, I_i) \rangle$
    $\mathcal{P} = \mathcal{P} \cup [y_i^v, \mathbf{q}_i]$
  end
  while inner loop do
    //Find a set of violating constraints
    foreach $i \in pos$ do
      $\{[v_i, \bar{\mathbf{q}}_i]\} \leftarrow \text{argmax}_{v,\bar{\mathbf{q}}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}_v(\bar{\mathbf{q}}, I_i) \rangle + \Delta([v, \bar{q}_0], y_i)$
      $\mathcal{S} = \mathcal{S} \cup \{[v_i, \bar{\mathbf{q}}_i]\}$
    end
    //Find a set of hard negative examples
    foreach $i \in neg$ do
      $\{[v_i, \bar{\mathbf{q}}_i]\} \leftarrow \text{argmax}_{v,\bar{\mathbf{q}}_i} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}_v(\bar{\mathbf{q}}_i, I_i) \rangle$
      $N = N \cup \{[v_i, \bar{\mathbf{q}}_i]\}$
    end
    $\boldsymbol{\theta} \leftarrow$ sgd ($\boldsymbol{\theta}, \mathcal{P}, \mathcal{S}, \mathcal{N}$); //update model
  end
end
```

*Optimization.* We solve (5) using our own implementation of stochastic gradient descent (SGD) with delayed constraint generation. The latent variables turn the optimization problem into a mixed integer program, solved using coordinate descent. Algorithm 1 describes the DPM-VOC+VP learning in detail. We start by initializing the model

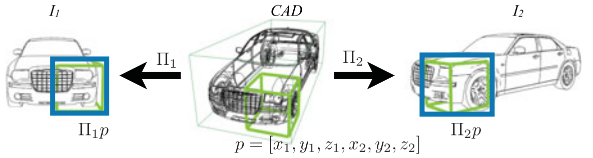


Fig. 4. 3D part parametrization for an example 3D CAD model (center). Corresponding projected part positions in two different views (left, right).

(line 1), and learning the root appearance terms $\alpha_{0,v}$ for each viewpoint component independently, using a standard SVM. The main part of the algorithm has an outer and inner loop. In the outer loop, the latent parts $\mathbf{q}_i$ are found for every positive training example $I_i$ (line 5), resulting in a set $\mathcal{P}$ of positive training examples. Then in an inner loop, the top $K$ ($K = 10$ in our experiments) active violating constraints $[v_i, \bar{\mathbf{q}}_i]$ are found for every positive training example $\{y_i, I_i\}$. This is the loss-augmented inference problem (line 10) and yields a current set of active constraints $\mathcal{S}$. We choose the violating constraints such that $\frac{A(\bar{q}_0 \cup y^b)}{A(\bar{q}_0 \cap y^b)} > 0.1$. Then, we search for negative examples from the negative labeled images (line 14), resulting in a set of "hard" negative examples $\mathcal{N}$. Finally the model parameters $\boldsymbol{\theta}$ are found by SGD (line 17).

### 3.4 DPM-3D-Constraints

The DPM-VOC+VP parameterizes part positions in 2D image space, independently across viewpoints. In this section we introduce the DPM-3D-Constraints that fundamentally changes the parameterization and works with parts in 3D. This way of modeling reflects the nature of the problem, observed are only 2D projections of what really are physical objects in a 3D world. Therefore, a parameterization in 3D appears both more meaningful and also should be beneficial for applications such as 3D object tracking or multi-view reconstruction.

Since annotated data is only available as 2D information, we use CAD models of the object classes of interest in addition to the annotated images. Being constructed of triangular surface meshes, 3D CAD models provide geometric descriptions of object class instances, lending themselves to 3D part parameterizations.

*Representation.* The DPM-3D-Constraints has the same graph structure as the DPM-VOC+VP, (see Fig. 3b). The difference is for every discrete viewpoint component there is a perspective projection matrix $\Pi_v$ that connects the 3D parameterization of parts with the 2D part placement observation (see Fig. 4).

For every part $p$ we need to specify the appearance (unary factor), and 2D part placement (pairwise factor). We use the setup of [25] to generate a non-photorealistic, gradient-based renderings of 3D CAD models. The renderings are used to compute HOG features for each part $p$. From a 3D bounding cube $p_i$ of a part, with $q_i = \Pi_v p_i$[3] we denote the 2D BB obtained by projecting the part into the viewpoint $v$. Then, the appearance features of the part are computed from the projected BB $\boldsymbol{\psi}(p_i, v, I) := \boldsymbol{\psi}(\Pi_v p_i, I)$.

The pairwise factor acts on 3D parts and computes the relative placement in the projected space. For the 3D root $p_0$ and part $p_i$, the feature function of the DPM-VOC+VP is re-used, but after projections $\eta(p_i, p_0, v) = \eta(\Pi_v p_i, \Pi_v p_0)$. There are separate parameters $\beta_v$ for every viewpoint component $v$.

In summary, the score of a 3D object hypothesis $\mathbf{p}$ and viewpoint $v$ is

$$\begin{aligned}\langle \boldsymbol{\theta}_v, \boldsymbol{\psi}(\mathbf{p}, v, I)\rangle = &\sum_{i=0}^{P} \langle \alpha_{i,v}, \phi(\Pi_v p_i, I)\rangle \\ &+ \sum_{i=1}^{P} \langle \beta_{i,v}, \eta(\Pi_v p_i, \Pi_v p_0)\rangle.\end{aligned} \tag{7}$$

There are two main differences between DPM-3D-Constraints and DPM-VOC+VP. First, the 2D parts $q_i$ are observed as projections $\Pi_v$ of their 3D counterparts $p_i$. Second, the model establishes part correspondences between different viewpoints. That is for a CAD model for which multiple renderings from different viewpoints are available, the estimated parts will be in correspondence across the renderings. Fig. 3b, illustrates the model. The dotted lines emphasize the deterministic relation (projection $\Pi_v$) between the 3D parts $p_i$ and their 2D counterparts $q_i$.

*Inference.* The inference problem is the same as for DPM-VOC+VP. We solve for the MAP estimate $\text{argmax}_{v,\mathbf{q}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}(\mathbf{q}, v, I)\rangle$. The predicted BB and viewpoint is provided by the highest scoring mixture component.

*Learning.* The optimization problem, loss function $\Delta_{VOC+VP}$, and Algorithm 1 for the DPM-3D-Constraints is the same as for DPM-VOC+VP. Different is the use of CAD data. During learning (Algorithm 1, line 5) 3D part positions of multiple renderings of a CAD model from different viewpoints are inferred. We enforce these to be consistent in 3D (thus the name DPM-3D-Constraints).

The training data in the form of images and annotations are augmented with a set of 3D CAD models $\{y^{\circ}\}$ of the object class of interest. Both are needed. The non-synthetic examples contribute to a realistic appearance model and the CAD models are used to encode 3D object geometry. We found that learning an appearance model from CAD data alone is not expressive enough. Assume we are given a 3D instance $y^{\circ}$, then let $S(y^{\circ})$ denote the set of all projections of $y^{\circ}$. Further we know the precise viewpoint $v_i$ $\forall i \in S(y^{\circ})$. For a 3D instance the inference is coupled via the set of all its projections

$$p^* = \underset{p}{\text{argmax}} \sum_{i \in S(y^o)} \left\langle \boldsymbol{\theta}_{v_i}, \boldsymbol{\psi}(\Pi_{v_i} p, v_i, I_i)\right\rangle. \tag{8}$$

For part initialization, we use the same data-driven method of the DPM-VOC+VP, but now in 3D. First, we define a part to be a 3D cube with size equal to 10 percent of the largest object size. Second, we choose greedily $k$ non-overlapping part positions with maximal combined appearance score across views.

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 ones with visible area higher than 10 percent of the area of the projected 3D part cube.

[3] Although the projection in general results in an arbitrary 2D polygon, we use $q_i = \Pi_v p_i$ to denote the 2D BB surrounding it.

### 3.5 3D$^2$PM

In this section we describe the 3D$^2$PM model, a 3D DPM entirely defined in 3D space. The 3D$^2$PM defines a conditional distribution over 3D object hypotheses $\mathbf{p}$ and only implicitly, through marginalization, for 2D object hypotheses $\mathbf{q}$. While DPM-3D-Constraints uses a 3D part parameterization, it is still a mixture model with different mixture components for different viewpoints, being limited to discrete set of viewpoints. The 3D$^2$PM model is continuous in the viewpoint variable $v \in \mathcal{V}$.

*Representation.* Starting from DPM-3D-Constraints, two ingredients are needed to obtain a full 3D object model: a continuous appearance model, and a 3D part displacement distribution.

For the definition of the continuous unary factor we introduce a number of support views $v_k, k = 1, \dots, K$. For a given viewpoint $v$ we then define the unary factor to be the weighted combination

$$\langle \alpha_{i,v}, \boldsymbol{\phi}(\Pi_v p_i, I)\rangle = \sum_{k=1}^{K} w_k(v) \langle \alpha_{i,v_k}, \boldsymbol{\phi}(\Pi_k p_i, I)\rangle, \tag{9}$$

with $w_k(v)$ being a viewpoint dependent scalar. The parameters of this model are thus the collection of all unary factors for parts and support views $\alpha_{i,v_k}, i = 1, \dots, P, k = 1, \dots, K$. In practice, we choose the support views to be equally spaced in angular distance $\delta_v = \angle(v_k, v_{k-1})$ on the viewing circle. This appearance score interpolates for a viewpoint $v$ filters from neighboring viewpoints. We experiment with three different models that correspond to different interpolations (i.e., choices of $w_k$): (i) linear interpolation, (ii) exponential interpolation, and (iii) a discrete set of views. In (i) we set $w_k = 1 - \frac{\angle(v,v_k)}{\angle(v_{k-1},v_k)}$ for the two closest support views, and $w_k = 0$ for the rest. We refer to this model as 3D$^2$PM-C-Lin, as it uses linear interpolation scheme. In (ii) we set $w_k = \exp(-\angle^2(v, v_k))$ and refer to the model as 3D$^2$PM-C-Exp. Finally, in (iii) we set $w_k = \mathbf{1}_{v=v_k}$ and we refer to this model as 3D$^2$PM-D as it can output a discrete set of viewpoints only.

For a given part $p_i$ and root $p_0$, the pairwise factor scores the joint displacement, again using a Gaussian term, but different to previous models, in 3D $\langle \beta_i, \eta_i(p_0, p_i)\rangle \propto -\ln(\mathcal{N}(p_i|p_0, \mu_i, \Sigma_i))$. The pairwise parameters are the $\beta_i(p_0, p_i) = [\mu_{ix}, \mu_{iy}, \mu_{iz}, \sigma_{ix}, \sigma_{iy}, \sigma_{iz}]$ and the feature function computes $\eta_i(p_0, p_i) = -[dx, dy, dz, dx^2, dy^2, dz^2]$. This factor contains only six parameters per part, in contrast to the previous models where $4K$ displacement parameters per part are required.

To define the score for a 2D object hypothesis $\mathbf{q}$ in an arbitrary viewpoint $v$, the 3D part displacement distribution is projected to 2D. For an arbitrary viewpoint $v$ the 3D part displacement distribution is projected via a scaled orthographic projection $Q_{i,v}$. The resulting distribution is the marginal of the Gaussian under this projection. Therefore the mean $\mu_{i,v} = Q_{i,v}\mu_i$, and covariance $\Sigma_{i,v} = Q_{i,v}\Sigma_i Q_{i,v}^\top$ can be computed in closed form. The parameters of the pairwise factor in viewpoint $v$ can be computed from the 3D parameters by $\beta_{i,v} = [\mu^u_{i,v}, \mu^v_{i,v}, \sigma^u_{i,v}, \sigma^v_{i,v}, \sigma^{uv}_{i,v}]$. Analogously, the 2D displacement features are $\eta_i(\Pi_v p_0, \Pi_v p_i) = -[du, dv, du^2, dv^2, 2dudv]$.

In summary both factors define the score of a hypotheses $\mathbf{p}$ under viewpoint $v$ for an observation $I$

$$\begin{aligned}\langle \boldsymbol{\theta}, \boldsymbol{\psi}(\mathbf{p}, v, I)\rangle = &\sum_{i=0}^{P} \langle \alpha_{i,v}, \boldsymbol{\phi}(\Pi_v p_i, I)\rangle \\ &+ \sum_{i=1}^{P} \langle \beta_{i,v}, \eta(\Pi_v p_i, \Pi_v p_0)\rangle.\end{aligned} \tag{10}$$

For a given 3D model $y^\circ$, and its projected images $S(y^\circ)$, under viewpoints $v_j$ the score of the 3D object hypothesis $\mathbf{p}$ is

$$\langle \boldsymbol{\theta}, \boldsymbol{\psi}(\mathbf{p}, y^\circ)\rangle = \sum_{i=0}^{P} \langle \alpha_i, \boldsymbol{\phi}(p_i, y^\circ)\rangle + \sum_{i=1}^{P} \langle \beta_i, \eta(p_i, p_0)\rangle. \tag{11}$$

Here, $\langle \alpha_i, \psi(p_i, y^\circ)\rangle$ is a 3D unary term, defined as $\langle \alpha_i, \boldsymbol{\phi}(p_i, y^\circ)\rangle = \sum_{S(y^\circ)} \langle \alpha_{i,v_j}, \boldsymbol{\phi}(\Pi_{v_j} p_i, I_j)\rangle$. It accumulates the 2D unary terms for every part from all projected images of the 3D model.

The 3D$^2$PM model $\boldsymbol{\theta} = [\boldsymbol{\alpha}, \boldsymbol{\beta}]$, consists of the unary parameters of the support views $[\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_K]$, as well as the parameters of the 3D displacement distribution of each part $\boldsymbol{\beta} = [\beta_1, \dots, \beta_P]$. Note that the 3D part displacement distributions are independent of the viewpoint components, that is every $\beta_i \in \mathbb{R}^3$. Fig. 3c illustrates the 3D$^2$PM model. Note that the double circle on the viewpoint variable $v$ denotes that it is continuous. The 2D displacement parameters $\beta_{i,v}$ are obtained via projection from the 3D displacement parameters $\beta_i$, therefore they are denoted with an empty factor.

*Inference.* The 3D$^2$PM output are 2D or 3D object hypotheses. For an observed image $I$, we solve again for the MAP estimate which corresponds to the following optimization problem: $\mathbf{q}^*, v^* = \operatorname{argmax}_{v,\mathbf{p}} \langle \boldsymbol{\theta}_v, \boldsymbol{\psi}(\mathbf{p}, v, I)\rangle$. For the 3D$^2$PM-D the viewpoint variable is discrete and the inference is the same as for DPM-3D-Constraints.

The MAP inference problem for 3D$^2$PM-C is a continuous problem. In practice we choose, at test time, an arbitrarily fine viewpoint binning. After this discretization, we proceed with the same inference procedure as for 3D$^2$PM-D. Note that this is different from choosing a viewpoint discretization at training time. This model allows to estimate the viewpoint up to an arbitrary precision, only chosen at test time.

For a 3D example $y^\circ$, the inference problem is $\mathbf{p}^* = \operatorname{argmax}_{\mathbf{p}} \langle \boldsymbol{\theta}, \boldsymbol{\psi}(\mathbf{p}, y^\circ)\rangle$, that is a consistent output is required for all images of the same instance. First, all unary terms are computed by collecting evidence from all available image projections of $y^\circ$, then, the 3D distance transform can be used to solve for optimal part placements in 3D.

*Learning.* Real images and additionally 3D models are used for training. We assume that the training data comes with angular accurate viewpoint annotations $y^v \in [0^\circ, 360^\circ)$. The 3D$^2$PM and all of its variants are learned using the same regularized risk objective as the DPM-VOC+VP, described in Eq. (5). Again, the loss measures detection and viewpoint estimation, with the difference that the loss

TABLE 1
Comparison of Different Models in Terms of Part Parameterization, Appearance Model, Component Initialization and Training Loss

| model | parts | | appear. | init | loss |
|---|---|---|---|---|---|
| | pos. | displ. | | | |
| DPM-Hinge [1] | 2D | 2D | disc. | AR | hinge |
| DPM-VOC | 2D | 2D | disc. | AR | voc |
| DPM-VOC+VP | 2D | 2D | disc. | VP | vocvp |
| DPM-3D-Constraints | 3D | 2D | disc. | VP | vocvp |
| 3D$^2$PM | 3D | 3D | cont. | VP | vocvp |

reflects the continuous viewpoint estimate $\Delta_{VP}(y, \bar{y}) = \angle(y^v, \bar{y}^v)/180$. Note that the learning algorithm does not need to change, we use Algorithm 1 with the only modification in line 17. To obtain the gradients w.r.t. the 3D parameters $\beta$, we simply take the gradients via the projected parameters using the chain rule. Table 1 summarizes the qualitative differences among the different models.

*Rendering CAD models.* We use the non-photorealistic wireframe-like renderings of [62], using a perspective projection. Depending on the dataset, we render all the CAD models in $\{8, 12, 16, 18, 24, 36\}$ equally spaced viewpoints, independently from the viewpoint statistics of the given dataset.

## 4 Experiments

In this section, we thoroughly evaluate our models on various datasets measuring their performance in terms of 2D BB localization, viewpoint estimation, and, in the case of DPM-3D-Constraints and 3D$^2$PM, their ability to predict part that correspond across viewpoints. To that end, we follow the ordering of Section 3, and successively add 3D information to the models under consideration.

We start by analyzing the performance of our structured output learning framework in comparison to the standard DPM formulation [1], highlighting its ability to provide both better BB localization (DPM-VOC) and simultaneous viewpoint estimation (DPM-VOC+VP) (Section 4.2). Second, we examine the impact of parameterizing object parts in 3D object coordinates rather than in the 2D image plane (DPM-3D-Constraints and 3D$^2$PM), again for the task of 2D BB localization and viewpoint estimation (Section 4.3), demonstrating superior performance in comparison to both previous work in 3D object class modeling and the standard DPM [1]. Third, we leverage the ability of our 3D$^2$PM model to predict viewpoints of arbitrary granularity for fine-grained viewpoint estimation (Section 4.4), again outperforming prior work. And fourth, we apply DPM-3D-Constraints and 3D$^2$PM to the task of ultra-wide baseline matching, quantifying their ability to localize corresponding parts in multiple views of the same object (Section 4.5).

All experiments are conducted on publicly available standard benchmarks for the respective task (Section 4.1) and include extensive comparisons to previous work.

### 4.1 Data Sets

We commence with a brief overview of the five diverse datasets used in the experiments.

*Pascal VOC 2007.* The detection benchmark of the Pascal VOC suite [11] provides a challenging test bed for 2D bounding box localization of 20 object classes. It is considered challenging due to strong variations in object appearance, background clutter, and partial occlusion. The 2007 version [5] has emerged as the standard benchmark for object detection approaches.

*Pascal3D+.* Recently, 12 object classes of Pascal VOC 2012 have been enriched with additional viewpoint annotations [4] by fitting 3D CAD models to images in a semi-automatic procedure. The performance is measured in terms of simultaneous 2D BB localization and viewpoint estimation. A candidate detection can only qualify as a true positive if it satisfies both the VOC intersection-over-union criterion [11] and provides correct viewpoint class estimate. We refer to the joint metric as average viewpoint precision (AVP).

*3D object classes.* Introduced in 2007, the 3D Object Classes dataset [3] still constitutes the de-facto standard dataset for multi-view recognition (i.e., 2D BB localization and viewpoint estimation). It provides images of nine object classes taken under controlled conditions w.r.t. viewpoint (three discrete different camera distances, three elevations, and eight azimuth angles) but exhibiting considerable background clutter and challenging lighting variations. Viewpoint estimation on this dataset is typically phrased as an 8-class classification problem (one class per azimuth angle).

*EPFL multi-view cars.* This dataset [6] has been recorded in the course of a car exhibition, where cars are presented to the audience on rotating platforms. While it features only a single object per image, lighting conditions are challenging (bright lights lead to specularities and saturation effects). Viewpoint annotations are almost angle-accurate (derived from platform rotation speed) and provide for a challenging fine-grained viewpoint estimation benchmark.

*KITTI.* The KITTI dataset [2] has been recorded from a moving vehicle driving through the city of Karlsruhe. It comes with manual BB and viewpoint annotations derived from 3D Lidar scans. It is challenging due to significant amounts of occlusion.

### 4.2 Structured Output Learning

We first compare the performance of our structured output learning framework (DPM-VOC, DPM-VOC+VP, Section 3.3) to the standard DPM. We evaluate on the following three data sets: Pascal VOC 2007, 3D Object Classes, and Pascal3D+. In all experiments, we use images from the respective data sets for training, following the protocols established as part of the data sets.

*2D Bounding box localization.* Table 2 gives results for 2D BB localization on the Pascal VOC 2007 dataset, according to the Pascal criterion, reporting per-class average precision (AP). It compares our DPM-VOC (row 2) to the DPM-Hinge [59] (row 1) and to the multi-kernel learning approach of Vedaldi et al. [63] (row 3), both of which are considered to be among the state-of-the-art on this data set. We first observe that DPM-VOC outperforms DPM-Hinge on 18 of 20 classes, and [63] on eight classes. While the relative performance difference of 1.1 percent on average (31.4 percent AP vs. 30.3 percent AP) to DPM-Hinge is moderate in terms of numbers, it is consistent and speaks in favor of the structured loss over the standard hinge loss. In comparison to

TABLE 2
2D BB Localization Performance on Pascal VOC 2007 [5], Comparing our DPM-VOC to DPM-Hinge and [63]

| 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 [63] | **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** |

*Note that [63] uses a kernel combination approach that makes use of multiple complementary image features.*

[63] (32.1 percent AP), DPM-VOC loses only 0.7 percent while the DPM-Hinge has 1.8 percent lower AP. We note that [63] exploits a variety of different features for performance, while the DPM-VOC and DPM-Hinge use HOG features only.

Fig. 5 (left) gives the corresponding results for nine 3D object classes, comparing DPM-Hinge (col. 1), DPM-Hinge-VP (col. 2), and DPM-VOC+VP (col. 3), 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 five of nine classes. While the average improvement is again moderate (performance increases from 88.0 percent over 88.4 percent to 88.7 percent AP), it confirms the benefit of the structured output objective, compared to the classification one.

*Viewpoint estimation.* Fig. 5 (right) gives results for viewpoint estimation, phrased as a classification problem, distinguishing among eight distinct azimuth angle classes. In line with previous work [3], [29], we report the mean precision in pose estimation (MPPE) on true positive detections according to the Pascal criterion (equivalent to the average over the diagonal of the confusion matrix). While there is an explicit association between mixture components and viewpoints for DPM-Hinge-VP and DPM-VOC+VP, we let the DPM-Hinge predict the most likely viewpoint by collecting votes from training example annotations for each component.

Clearly, the explicit association between viewpoints and mixture components already helps significantly (74.7 percent DPM-Hinge-VP vs. 55.8 percent DPM-Hinge), but we achieve a further boost by 12.4 percent in performance by applying a structured rather than hinge-loss (87.1 percent DPM-VOC+VP vs 74.7 percent DPM-Hinge-VP). A nice side effect is that training becomes considerably faster when fixing the mixture component assignments.

*Simultaneous BB localization and VP estimation.* So far, we have evaluated viewpoint estimation under rather special conditions. We have considered with 3D Object Classes a dataset for which 2D BB localization performance has essentially saturated beyond 95 percent AP for many classes. Then, we have evaluated viewpoint estimation entirely separately from 2D BB localization, on successful detections. While this is in line with standard evaluation procedures and prior work, it seems artificial for higher level applications, such as scene-understanding, or object tracking which require to solve both tasks simultaneously.

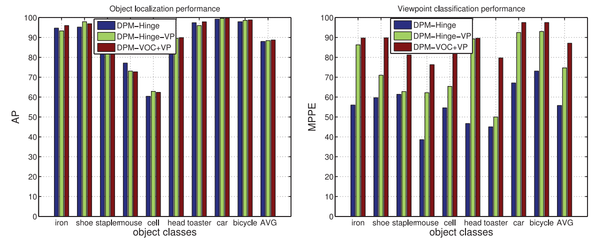


Fig. 5. 2D bounding box localization (left) and viewpoint estimation (right) results on nine 3D Object classes [3].

We hence turn to the recently proposed Pascal3D+ dataset that is both highly challenging in terms of 2D BB localization and comes with viewpoint annotations that allow to evaluate AVP (Section 4.1) in four different granularities (4, 8, 16, and 24 viewpoint classes). As baselines we again use DPM-Hinge, as well as the VDPM introduced in [4]. The VDPM is a viewpoint initialized DPM-Hinge (similarly to DPM-Hinge-VP), except that [4] flips the viewpoint components, resulting in twice as many components compared to DPM-VOC+VP.

Table 3 provides the corresponding AVP results and also gives separate 2D BB localization AP results as a reference. In terms of AVP, DPM-VOC+VP (24.5, 22.2, 17.9, and 14.4 percent for the four different viewpoint granularities) outperforms both the VDPM (19.5, 18.7, 15.6, 12.1 percent) and the DPM-Hinge (21.1, 13.2, 7.5, 3.0 percent) by large margins, for all viewpoint granularities (it improves over the VDPM by 5.0, 3.5, 2.3, and 2.3 percent respectively, and over the DPM-Hinge by 3.4, 9.0, 10.4 and 11.4 percent). Interestingly, DPM-VOC+VP can better deal with opposing object viewpoints than VDPM, since it explicitly incorporates the viewpoint loss.

In terms of pure 2D BB localization, our DPM-VOC+VP with 27.5, 28.8, 29.0, 28.2 percent outperforms the DPM-Hinge (28.2, 26.8, 25.4, 23.5 percent) on three of four viewpoint granularities. Compared to VDPM (26.8, 29.9, 30.0, 29.5 percent), DPM-VOC+VP is slightly worse (0.7 percent on average), which can be attributed to the fact that VDPM flips the viewpoint components, thus effectively having two components per viewpoint.

### 4.3 3D Object Class Representations

In the previous section, we confirmed improvements from the structured output learning framework for 2D BB localization and viewpoint estimation over the standard DPM. Here, we analyze the impact of adding 3D information, by first introducing a 3D part parameterization (DPM-3D-Constraints) and then adding 3D part displacement and continuous appearance models (3D$^2$PM). Experiments are conducted on 3D Object Classes, KITTI and Pascal VOC 2007.

In addition, we examine the effect of adding synthetic data in the form of rendered 3D CAD models (Section 3.4) to the respective training sets of real-world images, resulting in two different training data settings: (i) real data only,

TABLE 3
The Results of DPM-Hinge, VDPM and DPM-VOC+VP Are Shown

| AP/AVP | **aeroplane** | **bicycle** | **boat** | **bus** | **car** | **chair** | **diningtable** | **motorbike** | **sofa** | **train** | **tvmonitor** | **Avg.** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **DPM-Hinge-4V** | 35.3/20.3 | **47.8**/26.2 | **3.6**/**2.0** | **52.3**/**49.8** | 35.1/24.8 | **13.9**/**6.9** | **9.9**/**9.5** | **39.8**/23.0 | 10.7/10.3 | **26.7**/**23.9** | 34.9/**34.8** | **28.2**/21.1 |
| **DPM-Hinge-8V** | 35.6/3.9 | 45.7/6.4 | **6.3**/**1.2** | 48.1/**44.9** | 38.1/17.0 | **14.2**/3.6 | 9.2/4.0 | 34.3/5.9 | 5.6/4.4 | 24.2/**20.8** | 33.3/**32.7** | 26.8/13.2 |
| **DPM-Hinge-16V** | 33.4/1.0 | 43.1/1.0 | 3.9/0.3 | 44.9/26.7 | 36.7/6.9 | **15.3**/1.5 | 5.8/1.8 | 32.7/1.0 | 11.0/6.1 | 21.8/16.1 | 30.5/20.0 | 25.4/7.5 |
| **DPM-Hinge-24V** | 28.7/0.3 | 41.1/0.4 | 3.7/0.3 | 38.8/4.9 | 35.6/2.6 | **13.0**/0.8 | 8.2/2.0 | 30.1/1.0 | 10.1/4.9 | 21.3/6.2 | 28.1/9.0 | 23.5/3.0 |
| **VDPM - 4V** | 40.0/34.6 | 45.2/41.7 | 3.0/1.5 | 49.3/26.1 | 37.2/20.2 | 11.1/6.8 | 7.2/3.1 | 33.0/30.4 | 6.8/5.1 | 26.4/10.7 | **35.9**/34.7 | 26.8/19.5 |
| **VDPM - 8V** | 39.8/23.4 | 47.3/36.5 | 5.8/1.0 | 50.2/35.5 | 37.3/23.5 | 11.4/5.8 | **10.2**/**3.6** | 36.6/25.1 | **16.0**/**12.5** | **28.7**/10.9 | **36.3**/27.4 | **29.9**/18.7 |
| **VDPM - 16V** | **43.6**/15.4 | **46.5**/18.4 | **6.2**/0.5 | 54.6/46.9 | 36.6/18.1 | 12.8/6.0 | **7.6**/2.2 | 38.5/16.1 | **16.2**/**10.0** | **31.5**/**22.1** | **35.6**/16.3 | **30.0**/15.6 |
| **VDPM - 24V** | **42.2**/8.0 | 44.4/14.3 | **6.0**/0.3 | 53.7/39.2 | 36.3/13.7 | 12.6/4.4 | **11.1**/**3.6** | 35.5/10.1 | **17.0**/**8.2** | **32.6**/20.0 | **33.6**/11.2 | **29.5**/12.1 |
| **DPM-VOC+VP - 4V** | **43.8**/**39.4** | 47.0/**43.9** | 0.5/0.3 | 51.7/49.1 | **46.3**/**37.6** | 9.2/6.1 | 5.7/3.0 | 34.7/**32.2** | **13.3**/**11.8** | 17.4/12.5 | 33.4/33.2 | 27.5/**24.5** |
| **DPM-VOC+VP - 8V** | **42.0**/**29.7** | **49.8**/**42.6** | 0.9/0.4 | **52.0**/39.5 | **47.9**/**36.8** | 11.3/**9.4** | 5.3/2.6 | **39.8**/**32.9** | 13.5/11.0 | 21.4/10.3 | 33.1/28.6 | 28.8/**22.2** |
| **DPM-VOC+VP - 16V** | 39.3/**17.0** | 46.3/**24.7** | 2.6/**1.0** | **55.3**/**49.0** | **46.0**/**30.1** | 10.4/**6.6** | 7.5/**3.0** | **39.5**/**17.2** | 12.7/7.7 | 28.5/20.4 | 30.7/**20.2** | 29.0/**17.9** |
| **DPM-VOC+VP - 24V** | 37.7/**10.6** | **45.9**/**16.7** | 5.6/**2.2** | **55.2**/**43.5** | **42.9**/**25.4** | 9.1/**4.4** | 7.6/2.3 | **35.7**/**11.3** | 11.5/4.9 | 31.1/**22.4** | 27.6/**14.4** | 28.2/**14.4** |

*The first number indicates the average precision (AP) for detection and the second number shows the AVP for joint object detection and pose estimation.*

and (ii) mixed data (real and synthetic). Please note that both DPM-3D-Constraints and 3D$^2$PM always employ 3D CAD models for establishing a 3D coordinate system, irrespective of whether synthetic images are used for training appearance models.

*3D object classes.* Table 4 compares the DPM-VOC+VP, DPM-3D-Constraints, and 3D$^2$PM with state-of-the-art results on 3D object classes [3], distinguishing 2D and 3D object class representations. We make the following observations. First, DPM-VOC+VP (91.3 percent AP, 91.6 percent MPPE) outperforms all other methods on average (last row) as well as on six of nine classes. It outperforms the next best prior result of 82.3 percent AP and 81.3 percent MPPE obtained by the aspect layout model (ALM) [54] by 9.0 and 10.3 percent respectively, despite the ALM making use of additional human annotation in the form of aspect layout parts. Second, the top performance of DPM-VOC+VP is almost matched by both of our 3D object class representations, DPM-3D-Constraints (89.6 percent AP, 91.1 percent MPPE) and 3D$^2$PM (90.4 percent AP, 89.4 percent MPPE). This is remarkable since the 3D representations put additional (3D) constraints on the learned model, while DPM-VOC+VP is only bound by the combined localization and viewpoint loss, directly optimizing for the task at hand without any additional constraints. And third, we see that our models also compare favorably to prior work that has specialized on certain object classes, such as cars. Specifically, 3D$^2$PM outperforms the voting-based approach of [28] (99.2 percent AP, 85.3 percent MPPE) which relies on 3D reconstructions of the object class of interest as training data.

*KITTI.* Table 6 provides 2D BB localization and viewpoint estimation results on the challenging KITTI dataset for our models, trained from either purely real or mixed training data. We split the dataset into three equal sets, used for training, validation and testing. Starting with the real data setting, we observe that all our models consistently outperform the DPM-Hinge: the improvements in average AP range from 0.6 percent (3D$^2$PM) over 2.2 percent (DPM-3D-Constraints) to 5.8 percent (DPM-VOC+VP) and in MPPE from as much as 22.3 percent (3D$^2$PM) over 26.9 percent (DPM-3D-Constraints) to 29.3 percent (DPM-VOC+VP). Comparing our different models, the DPM-VOC+VP performs best (47.7 percent AP, 54.3 percent MPPE), followed by DPM-3D-Constraints (44.1 percent AP, 51.9 percent MPPE) and 3D$^2$PM (42.5 percent AP and 47.3 percent MPPE)—it seems that the added expressiveness of our 3D models DPM-3D-Constraints and 3D$^2$PM comes at a (moderate) cost w.r.t. performance, which we attribute to occlusion. 3D$^2$PM performs worse on medium to highly occluded objects, compared to DPM-VOC+VP. On the 0-20 percent occlusion level, 3D$^2$PM (79.2 percent) achieves 0.3 percent better performance than DPM-VOC+VP (78.9 percent), but on the rest of the occlusion levels it is consistently worse (e.g. on 60-80 percent DPM-VOC+VP is better by 3.0 percent).

Adding synthetic training images improves the performance of our models mostly for viewpoint estimation: DPM-3D-Constraints improves by 2.4 percent, from 51.9 to 54.3 percent MPPE, and 3D$^2$PM from 47.3 to 47.7 percent MPPE. For 2D BB localization, only 3D$^2$PM improves by

TABLE 4
Comparison to State-of-the-Art in 2D BB Localization and Viewpoint Estimation on 3D Object Classes [3]

| AP/<br>MPPE | 2D Models | | | | 3D Models | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | DPM-VOC<br>+VP | Lopez<br>[29] | Bao<br>[50] | Payet<br>[27] | 3D$^2$PM | DPM-3D-<br>Constraints | ALM [54] | Yoruk<br>[53] | Liebelt [24] | Zia<br>[64] | Glasner<br>[28] |
| car | **99.9/97.9** | 96.0/87.9 | 98.0/95.3 | -/86.1 | 99.8/97.5 | 99.4/97.5 | 98.3/93.1 | 93.3/73.0 | 76.7/70.0 | 90.4/84.0 | 99.2/85.3 |
| bicycle | **98.8/97.5** | 91.0/89.9 | 93.1/92.3 | -/80.8 | 96.6/96.4 | 95.8/96.1 | 93.8/90.1 | -/- | 69.8/75.5 | -/- | -/- |
| iron | **98.1/94.2** | 53.0/90.8 | 82.5/89.8 | -/- | 97.2/93.1 | 97.7/93.6 | 82.2/86.0 | -/- | -/- | -/- | -/- |
| shoe | **98.8/97.6** | 78.0/89.3 | 85.5/88.0 | -/- | 98.3/95.8 | 97.9/96.1 | 84.1/86.6 | -/- | -/- | -/- | -/- |
| stapler | **89.8/92.6** | 32.0/79.3 | 70.2/73.9 | -/- | 88.4/86.9 | 86.4/89.1 | 70.5/73.2 | -/- | -/- | -/- | -/- |
| mouse | **77.4**/82.0 | 41.0/66.4 | 54.5/72.0 | -/- | 74.9/83.5 | 77.0/**88.3** | 52.2/69.8 | -/- | -/- | -/- | -/- |
| cell. | 71.4/90.7 | 43.0/75.4 | **81.0**/86.0 | -/- | 70.1/91.2 | 67.4/**92.7** | 80.2/86.3 | -/- | -/- | -/- | -/- |
| head | **90.9**/**90.7** | 76.0/77.4 | -/- | -/- | 92.5/88.7 | 88.5/88.7 | -/- | -/- | -/- | -/- | -/- |
| toaster | 97.0/**81.6** | 54.0/56.9 | **98.2**/70.3 | -/- | 95.4/71.9 | 96.4/78.1 | 97.5/65.4 | -/- | -/- | -/- | -/- |
| avg | **91.3/91.6** | 62.7/79.2 | 83.0/83.5 | -/ - | 90.4/89.4 | 89.6/91.1 | 82.3/81.3 | -/ - | -/- | -/- | -/- |

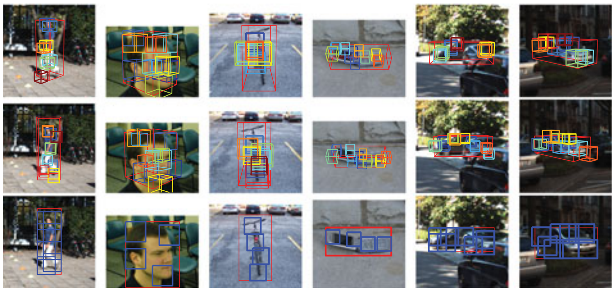


Fig. 6. Qualitative results on KITTI and 3D object classes. Corresponding part detections (for a given class) are color coded. 3D$^2$PM (first row), DPM-3D-Constraints (second row) and DPM-VOC+VP (third row).

1.5 percent from 42.5 to 44.0 percent AP, while the other models lose performance (DPM-VOC+VP loses 0.5 percent AP, DPM-3D-Constraints loses 1.7 percent). We attribute this mixed behavior to the fact that synthetic training images come with perfect, angular accurate viewpoint annotations (improving viewpoint estimation), but often deviate from real-world training images in terms of appearance, at least for the chosen type of edge-based rendering—we leave improving the rendering quality for future work. Fig. 6 shows qualitative results on KITTI and 3D object classes.

Table 5 shows the results in terms of AP and AOS (average orientation similarity) [2], now on the KITTI testing set [2]. DPM-VOC+VP (39.3 percent), DPM-3D-Constraints (35.4 percent) and 3D$^2$PM (36.7 percent) outperform the DPM-Hinge (34.4 percent) across all the classes.

*Pascal VOC 2007.* So far, we have compared 3D$^2$PM only to 3D object models on the 3D object classes dataset. Next, we want to compare its performance to 3D object models, now on challenging datasets. 3D object models traditionally have issues when performing object detection on challenging benchmarks due to the large pose, shape, occlusion, and size variation that these benchmarks exhibit. To that end, we compare 3D$^2$PM on the Pascal VOC 2007 dataset to the 3D voting scheme in [28], as it is the only 3D object model we have found that reports performance numbers on challenging benchmarks. 3D$^2$PM-D achieves 62.2 percent AP on the car class outperforming the previous best 3D Object Model result of 32 percent AP of [28] by a large margin, while it is comparable to DPM-3D-Constraints (63.1 percent) and slightly worse than DPM-VOC (65.7 percent).

## 4.4 3D Deformations and Continuous Appearance

While accurate 2D BB localization and viewpoint classification into coarse classes can be achieved with a purely view-based 2D (DPM-VOC+VP, Section 4.2) or 3D (DPM-3D-Constraints, Section 4.3) object class representation, estimating viewpoint on a finer level of granularity demands a proper 3D object class model with 3D deformations and continuous appearance, such as 3D$^2$PM. In this section, we hence highlight the ability of our 3D$^2$PM to predict viewpoint up to arbitrary granularity. To that end, we use the EPFL Multi-view cars dataset (Section 4.1), due to its angle-accurate viewpoint annotations and uniform sampling of the viewing circle.

TABLE 5
2D BB Localization and Viewpoint Estimation on KITTI Testing [2]

| AP/ AOS | DPM-VOC+VP | DPM-3D-Const | 3D$^2$PM | DPM-Hinge |
|---|---|---|---|---|
| car | 48.8/46.5 | 42.2/40.1 | 45.6/42.9 | 41.0/- |
| ped. | 40.4/35.7 | 36.6/29.6 | 37.4/30.7 | 34.8/- |
| cycl. | 28.2/21.6 | 27.5/21.1 | 27.1/20.9 | 27.3/- |
| avg | 39.3/34.6 | 35.4/30.3 | 36.7/31.5 | 34.4/- |

### 4.4.1 Arbitrarily Fine Viewpoint Estimation

In order to assess the ability of our 3D$^2$PM models to generate viewpoint estimates of arbitrarily fine granularity, we train 3D$^2$PM-C with a varying number of $k \in \{8, 12, 16, 18, 36\}$ support views, interpolating to a varying number of predicted views of increasing resolution $d \in \{45°, 30°, 22.5°, 20°, 10°, 8°, 5°\}$. Fig. 7 plots the corresponding results for 3D$^2$PM-C-Lin (left) and 3D$^2$PM-C-Exp (right) as surfaces of MAE over $k$ and $d$.

For both models, we observe that both, increasing $k$ for fixed $d$ and decreasing $d$ for fixed $k$, in fact results in lower angular error in most cases, highlighting the benefit of the 3D continuous representation. The respective minima are attained at $k = 36$, $d = 5°$ (4.62 degrees MAE for 3D$^2$PM-C-Lin and 4.70 degrees for 3D$^2$PM-C-Exp), approaching the dataset viewpoint label noise.

### 4.4.2 Comparison to State-of-the-Art

Table 7 reports MAE for our 3D$^2$PM models at 5 degree resolution comparing to state-of-the-art. The 3D$^2$PM-D, 3D$^2$PM-C-Lin, and 3D$^2$PM-C-Exp models with $k = 8$ achieve 12.89, 12.43 and 12.63 degree MAE outperforming by almost 12 degree the best published result of 24.8 degree of [28].

Table 8 gives a comparison to prior results that had been measured in terms of 2D BB localization (AP) and viewpoint

TABLE 6
2D BB Localization and Viewpoint Estimation on KITTI [2]

| AP/MPPE | real | | | mixed | | | baseline |
|---|---|---|---|---|---|---|---|
| | DPM-VOC+VP | DPM-3D-Constr. | 3D$^2$PM | DPM-VOC+VP | DPM-3D-Constr. | 3D$^2$PM | DPM-Hinge |
| car | **63.0**/**73.6** | 61.6/70.7 | 60.3/63.2 | 61.4/70.8 | 60.8/71.3 | 61.3/65.6 | 60.5/46.1 |
| pedestrian | 43.7/**46.6** | 38.0/31.9 | 36.1/40.0 | **43.9**/45.4 | 35.9/45.6 | 38.9/41.3 | 36.2/22.9 |
| cyclist | **36.5**/42.6 | 32.7/**53.0** | 31.1/38.8 | 36.3/46.8 | 30.4/45.9 | 31.8/36.1 | 28.9/6.0 |
| AVG | **47.7**/54.3 | 44.1/51.9 | 42.5/47.3 | 47.2/54.3 | 42.4/**54.3** | 44.0/47.7 | 41.9/25.0 |

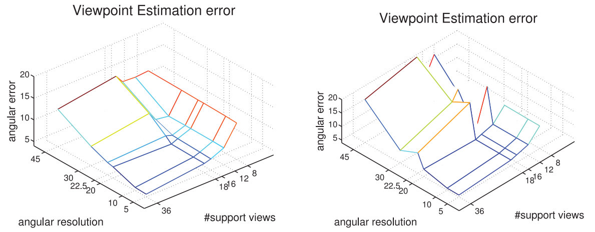


Fig. 7. Fine viewpoint estimation performance (in MAE) using linear (left) and exponential interpolation (right).

TABLE 7
Fine Viewpoint Estimation on EPFL [6]

| MAE | Glasner [28] | 3D$^2$PM-D | 3D$^2$PM-C-Lin | 3D$^2$PM-C-Exp |
|---|---|---|---|---|
| 8 bins | 24.80 | 12.89 | **12.43** | 12.63 |
| 12 bins | - | 7.99 | **7.89** | 7.99 |
| 16 bins | - | 7.00 | **6.59** | 6.77 |
| 18 bins | - | 6.29 | **6.15** | 6.15 |
| 36 bins | - | 4.74 | **4.62** | 4.70 |

estimation performance (MPPE) rather than MAE (note that MPPE is measured according to the respective number of support views and is not comparable across table rows). We observe that our models outperform prior results in AP and MPPE by significant margins. 3D$^2$PM-C-Lin (99.7 percent AP, 80.6 percent MPPE) performs best on average, outperforming [29] (91.0 percent AP, 73.7 percent MPPE) by 8.7 and 6.9 percent for eight support views, and by 1.8 and 7.5 percent for 16 views, respectively. Interpolation (3D$^2$PM-C-Lin and 3D$^2$PM-C-Exp) consistently improves performance by around 2-3 percent over 3D$^2$PM-D in terms of MPPE, and 3D$^2$PM-C-Lin is around 1-2 percent better than 3D$^2$PM-C-Exp on average.

### 4.4.3 *Coarse-to-Fine Viewpoint Inference*

As we go towards arbitrarily fine viewpoint estimation with 3D$^2$PM-C, we increase the number of model evaluations for a given position and viewpoint (atomic operation), increasing inference complexity. We thus propose a coarse-to-fine inference scheme that reduces the number of operations while not sacrificing too much performance. It uses a greedy binary search to recursively partition the space of candidate viewpoints considered.

Table 9 gives results at 5 degrees on EPFL, comparing 3D$^2$PM-C with $k = 36$ and full inference (row 1) to the same model with coarse-to-fine inference (row 2), starting with 12 views at the coarse level, and two reference models with $k \in 12, 16$. While achieving almost 5 times faster runtime (0.48 $\times 10^{10}$ vs. 2.2 $\times 10^{10}$ atomic operations), we obtain comparable AP (99.0 vs. 99.9 percent AP) and only slightly worse MAE (7.0 vs. 4.62 degree MAE), and comparable performance to the reference models, but at much lower computational cost. More sophisticated methodologies for approximate inference, such as Branch and Rank [65], could further improve run-time.

TABLE 9
Detection (AP) and vp. Estimation (MAE); Full vs. Coarse-to-Fine Inference

| AP / MAE | at 5° | #atomic operations |
|---|---|---|
| 3D$^2$PM-C vp36 full | 99.9/4.62 | 2.20 $\times 10^{10}$ |
| 3D$^2$PM-C vp36 coarse to fine | 99.0/7.00 | 0.48 $\times 10^{10}$ |
| 3D$^2$PM vp12 | 99.6/7.89 | 2.20 $\times 10^{10}$ |
| 3D$^2$PM vp16 | 99.8/6.59 | 2.20 $\times 10^{10}$ |

## 4.5 3D Part Correspondences

Lastly, we leverage the 3D nature of DPM-3D-Constraints and 3D$^2$PM to match parts across different viewpoints, as it is required in multi-view scene understanding or object tracking. In order to quantify this ability, we perform ultra-wide baseline matching as established by [64], and measure how often a fundamental matrix relating two views of the same object can be estimated from putative part correspondences.

Table 10 compares DPM-3D-Constraints and 3D$^2$PM with 12 and 20 parts, respectively, to raw SIFT matches and [64], for baselines from 45 to 180 degrees. We observe that both DPM-3D-Constraints (57.1 percent) and 3D$^2$PM (66.4 percent) with 20 parts outperform the prior result of [64] (52.0 percent) by considerable margins of 5.1 and 14.4 percent, respectively.

# 5 Conclusion

This paper extends the DPM [1] to include viewpoint and 3D geometry information, thus bringing the world of 2D object detectors and 3D object representations closer. By adding 3D geometry information on three different levels (viewpoints, part parameterization and part distributions), in this work we have provided a palette of object detectors, which gradually and successfully introduce object geometry into the DPM. The 3D$^2$PM extends the DPM to a full 3D object model. It leverages 3D information from CAD data, performing viewpoint estimation at arbitrarily fine granularity.

TABLE 8
2D BB Localization (AP) and Viewpoint Estimation (MPPE [29]) on EPFL [6]

| AP/MPPE | 3D$^2$PM-D | 3D$^2$PM-C-Lin | 3D$^2$PM-C-Exp | ALM [54] | Ozuy.[6] | Lopez [29] |
|---|---|---|---|---|---|---|
| 8 bins | **99.8**/77.6 | 99.7/**80.6** | 98.8/79.4 | -/- | -/- | 91.0/73.7 |
| 12 bins | 98.9/79.0 | **99.6**/**83.1** | 99.5/81.1 | -/- | -/- | - / - |
| 16 bins | 99.8/70.8 | **99.8**/73.5 | 99.6/**74.0** | 98.1/56.6 | 85.0/41.6 | 97.0/66.0 |
| 18 bins | 99.8/72.1 | 99.8/**75.0** | **99.9**/73.8 | -/- | -/- | -/- |
| 36 bins | 99.9/52.7 | **99.9**/**55.9** | 99.7/54.5 | -/- | -/- | -/- |

TABLE 10
Ultra-Wide Baseline Matching Performance, Measured by the Fraction of Correctly Estimated Fundamental Matrices

| Azim. | SIFT | [64] | DPM-3D-C. 12 | DPM-3D-C. 20 | $3D^2PM$ 12 | $3D^2PM$ 20 |
|---|---|---|---|---|---|---|
| 45 degree | 2.0% | 55.0% | 49.1% | 54.7% | 47.2% | 58.5% |
| 90 degree | 0.0% | 60.0% | 42.9% | 51.4% | 54.3% | 77.1% |
| 135 degree | 0.0% | 52.0% | 55.2% | 51.7% | 44.8% | 58.6% |
| 180 degree | 0.0% | 41.0% | 52.9% | 70.6% | 70.6% | 70.6% |
| AVG | 0.5% | 52.0% | 50.0% | 57.1% | 54.2% | 66.4% |

In an extensive experimental study on several datasets with varying level of difficulty, and on several different classes we have shown that the presented models achieve state-of-the-art performance in terms of viewpoint estimation and ultra-wide baseline part matching, confirming the ability to deliver expressive object hypotheses. Therefore, the models presented in this paper take a step forward towards bridging the gap between object detection and higher level tasks like scene understanding and 3D object tracking.

## References

[1] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. Int. Conf. Comput. Vision Pattern Recog.*, 2012, pp. 3354–3361.

[3] S. Savarese and L. Fei-Fei, "3D generic object categorization, localization and pose estimation," in *Proc. IEEE 11th Int. Conf. Comput. Vision*, 2007, pp. 1–8.

[4] Y. Xiang, R. Mottaghi, and S. Savarese, "Beyond Pascal: A benchmark for 3D object detection in the wild," in *Proc. IEEE Winter Conf. Appl. Comput. Vision*, 2014, pp. 75–82.

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL VOC 2007 Results," [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html, 2007.

[6] M. Ozuysal, V. Lepetit, and P. Fua, "Pose estimation for category specific multiview object localization," in *Proc. Proc. Int. Conf. Comput. Vision Pattern Recog.*, 2009, pp. 778–785.

[7] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. Int. Conf. Comput. Vision Pattern Recog.*, 2003, pp. II-264–II-271.

[8] B. Leibe, A. Leonardis, and B. Schiele, "An implicit shape model for combined object categorization and segmentation, " in *Toward Category-Level Object Recognition*. New York, NY, USA: Springer, 2006.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. Int. Conf. Comput. Vision Pattern Recog.*, 2014, pp. 580–587.

[10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," presented at the 2nd Int. Conf. Learning Representations, Banff, Canada, 2014.

[11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vision*, vol. 88, pp. 303–338, 2010.

[12] D. Marr and H. K. Nishihara, "Representation and recognition of the spatial organization of three-dimensional shapes," *Proc. R. Soc. Lond. B, Biol. Sci.*, vol. 200, pp. 269–294, 1978.

[13] R. A. Brooks, "Symbolic reasoning among 3-d models and 2-d images," *Artif. Intell.*, vol. 17, pp. 285–348, 1981.

[14] A. P. Pentland, "Perceptual organization and the representation of natural form," *Artif. Intell.*, vol. 28, pp. 293–331, 1986.

[15] D. G. Lowe, "Three-dimensional object recognition from single two-dimensional images," *Artif. Intell.*, vol. 31, pp. 355–395, 1987.

[16] L. Stark, A. Hoover, D. Goldgof, and K. Bowyer, "Function-based recognition from incomplete knowledge of shape," in *Proc. IEEE Workshop Qualitative Vision*, 1993, pp. 11–22.

[17] K. Green, D. Eggert, L. Stark, and K. Bowyer, "Generic recognition of articulated objects through reasoning about potential function," *Comput. Vision Image Understanding*, vol. 62, pp. 177–193, 1995.

[18] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," *Int. J. Comput. Vision*, vol. 80, pp. 3–15, 2008.

[19] A. Ess, B. Leibe, K. Schindler, and L. V. Gool, "Robust multi-person tracking from a mobile platform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1831–1846, Oct. 2009.

[20] C. Wojek, S. Roth, K. Schindler, and B. Schiele, "Monocular 3D scene modeling and inference: Understanding multi-object traffic scenes," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 467–481.

[21] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool, "Towards multi-view object class detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog.*, 2006, pp. 1589–1596.

[22] P. Yan, S. Khan, and M. Shah, "3D model based object class detection in an arbitrary view," in *Proc. 11th Int. Conf. Comput. Vision*, 2007, pp. 1–6.

[23] 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 *Proc. 12th Int. Conf. Comput. Vision*, 2009, pp. 213–220.

[24] J. Liebelt and C. Schmid, "Multi-view object class detection with a 3D geometric model," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2010, pp. 1688–1695.

[25] M. Stark, M. Goesele, and B. Schiele, "Back to the future: Learning shape models from 3D CAD data," in *Proc. Brit. Mach. Vision Conf.*, 2010, pp. 106.1–106.11.

[26] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler, "Detailed 3D representations for object recognition and modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 11, pp. 2608–2623, Nov. 2013.

[27] N. Payet and S. Todorovic, "From contours to 3D object detection and pose estimation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2011, pp. 983–990.

[28] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and pose estimation," in *Proc. IEEE Int. Conf. Comput. Vision*, 2011, pp. 1275–1282.

[29] R. J. Lopez-Sastre, T. Tuytelaars, and S. Savarese, "Deformable part models revisited: A performance evaluation for object category pose estimation," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2011, pp. 1052–1059.

[30] S. Bao and S. Savarese, "Semantic structure from motion," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2011, pp. 2025–2032.

[31] A. Geiger, C. Wojek, and R. Urtasun, "Joint 3D estimation of objects and scene layout," in *Proc. Neural Inf. Process. Syst. Conf.*, 2011, pp. 1467–1475.

[32] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun, "3D traffic scene understanding from movable platforms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 5, pp. 1012–1025, May 2014.

[33] R. Girshick, P. Felzenszwalb, and D. McAllester, "Object detection with grammar models," in *Proc. Neural Inf. Process. Syst. Conf.*, 2011, pp. 442–450.

[34] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, 2013, pp. 17–24.

[35] S. Fidler, R. Mottaghi, A. L. Yuille, and R. Urtasun, "Bottom-up segmentation for top-down detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2013, pp. 3294–3301.

[36] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vision*, vol. 104, pp. 154–171, 2013.

[37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2009, pp. 248–255.

[38] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "Sun database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2010, pp. 3485–3492.

[39] D. Lowe, "Distinctive image features from scale invariant keypoints," *Int. J. Comput. Vision*, vol. 60, pp. 90–110, 2004.

[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vision Pattern Recog.*, 2005, pp. 886–893.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst. Conf.*, 2012, pp. 1106–1114.

[42] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learning*, vol. 20, pp. 273–297, 1995.

[43] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2001, pp. I-511–I-518.

[44] L. Breiman, "Random forests," *Mach. Learning*, vol. 45, pp. 5–32, 2001.

[45] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2013, pp. 2553–2561.

[46] C. Gu and X. Ren, "Discriminative mixture-of-templates for viewpoint classification," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2010, pp. 408–421.

[47] S. Fidler, S. Dickinson, and R. Urtasun, "3D object detection and viewpoint estimation with a deformable 3D cuboid model," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2012, pp. 620–628.

[48] M. Hejrati and D. Ramanan, "Analyzing 3D objects in cluttered images," in *Proc. Adv. Neural Inf. Process. Syst. Conf.*, 2012, pp. 602–610.

[49] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2013, pp. 3286–3293.

[50] S. Y.-Z. Bao, Y. Xiang, and S. Savarese, "Object co-detection," in *Proc. Eur. Conf. Comput. Vision*, 2012, pp. 86–101.

[51] M. Arie-Nachimson and R. Basri, "Constructing implicit 3D shape models for pose estimation," in *Proc. Int. Conf. Comput. Vision*, 2009, pp. 1341–1348.

[52] M. Sun, B. Xu, G. Bradski, and S. Savarese, "Depth-encoded hough voting for joint object detection and shape recovery," in *Proc. Eur. Conf. Comput. Vision*, 2010, pp. 658–671.

[53] E. Yoruk and R. Vidal, "Efficient object localization and pose estimation with 3D wireframe models," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2013, pp. 538–545.

[54] Y. Xiang and S. Savarese, "Estimating the aspect layout of object categories," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2012, pp. 3410–3417.

[55] M. Hejrati and D. Ramanan, "Analysis by synthesis: 3D object recognition by object reconstruction," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2014, pp. 2449–2456.

[56] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic, "Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2014, pp. 3762–3769.

[57] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *Int. J. Comput. Vision*, vol. 61, pp. 55–79, 2005.

[58] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 2009, pp. 1014–1021.

[59] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. [Online]. Available: http://people.cs.uchicago.edu/pff/latent-release4/, 2010.

[60] C.-N. J. Yu and T. Joachims, "Learning structural SVMs with latent variables," in *Proc. 26th Annu. Int. Conf. Mach. Learning*, 2009, pp. 1169–1176.

[61] M. Blaschko and C. Lampert, "Learning to localize objects with structured output regression," in *Proc. 10th Eur. Conf. Comput. Vision*, 2008, pp. 2–15.

[62] M. Stark, J. Krause, B. Pepik, D. Meger, J. Little, B. Schiele, and D. Koller, "Fine-grained categorization for 3d scene understanding," in *Proc. Brit. Mach. Vision Conf.*, 2012, pp. 1–12.

[63] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *Proc. Int. Conf. Comput. Vision*, 2009, pp. 606–613.

[64] M. Z. Zia, M. Stark, K. Schindler, and B. Schiele, "Revisiting 3D geometric models for accurate object shape and pose," in *Proc. IEEE Int. Conf. Comput. Vision Workshops*, 2011, pp. 569–576.

[65] A. Lehmann, P. Gehler, and L. Van Gool, "Branch&rank: Non-linear object detection," in *Proc. Brit. Mach. Vision Conf.*, 2011, pp. 8.1–8.11.

**Bojan Pepik** received the Diploma degree in electrical engineering from UKIM Skopje, Macedonia, in 2008. Since then, he has worked as a researcher in the Digital Image Processing Team and at the Compute Science Department at FEIT UKIM, Macedonia. He joined the UdS Computer Science Graduate School in 2009. He is currently working toward the PhD degree in computer science in the Computer Vision and Multi-modal Computing group at the Max-Planck Institute for Informatics, Saarbrucken. His research interests include computer vision and machine learning. He is a student member of the IEEE.

**Michael Stark** received the Diploma degree in computer science from TU Darmstadt, Germany, in 2005, and the PhD degree from TU Darmstadt, Germany, in 2010. He has worked as a postdoc with the Max Planck Institute for Informatics in Saarbruecken, Germany, and Stanford University, Stanford, USA, where he was appointed as a Visiting Assistant Professor in 2012. Since then, he has been heading the research group for Visual Object Recognition and Scene Interpretation of the Max Planck Center for Visual Computing and Communication, first at Stanford, and since 2014, at the Max Planck Institute for Informatics in Saarbruecken, Germany. His research interests encompass computer vision and machine learning, focusing on scene understanding and 3D object class detection. He is a member of the IEEE.

**Peter Gehler** received the PhD degree from the Max Planck Institute for Biological Cybernetics. He is a Research Group leader at the Bernstein Center for Computational Neuroscience and the Max Planck Institute for Intelligent Systems in Tbingen, Germany. Previously, he was a postdoctoral researcher at ETH Zurich and a Research Scientist at the Max Planck Institute for Informatics. His main research interests include inference processes in artificial vision systems. He is a member of the IEEE.

**Bernt Schiele** received the master's degree in computer science from the University of Karlsruhe and INP Grenoble in 1994 and the PhD degree in computer vision from INP Grenoble in 1997. He was a postdoctoral associate and visiting assistant professor with MIT between 1997 and 2000. From 1999 until 2004, he was an assistant professor with ETH Zurich and, from 2004 to 2010, he was a full professor of computer science with TU Darmstadt. In 2010, he was appointed a scientific member of the Max Planck Society and the director at the Max Planck Institute for Informatics. Since 2010, he has also been a professor at Saarland University. His main interests include computer vision, perceptual computing, statistical learning methods, wearable computers, and integration of multimodal sensor data. He is particularly interested in developing methods which work under real-world conditions. He is a member of the IEEE.