3D^{2}PM – 3D Deformable Part Models

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

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

Abstract. As objects are inherently 3-dimensional, 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, 2D feature-based models are the predominant paradigm in object recognition today. While such models have shown competitive bounding box (BB) detection performance, they are clearly limited in their capability of fine-grained reasoning in 3D or continuous viewpoint estimation as required for advanced tasks such as 3D scene understanding. This work extends the deformable part model [1] to a 3D object model. It consists of multiple parts modeled in 3D and a continuous appearance model. As a result, the model generalizes beyond BB oriented object detection and can be jointly optimized in a discriminative fashion for object detection and viewpoint estimation. Our 3D Deformable Part Model (3D^{2}PM) leverages on CAD data of the object class, as a 3D geometry proxy.

1 Introduction

In the early days, 3D representations of objects and entire scenes were considered the holy grail [2–5]. 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 [6, 7]. However, 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 representations of object classes based on robust local features and statistical learning techniques [8–10, 1].

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 [11–13]. This has also fuelled the development of multi-view recognition methods [14–23], providing viewpoint estimates as additional cue for scene-level reasoning [24]. Most approaches, however, are still either limited with respect to the degree of 3D modelling, or can not provide competitive performance in terms of 2D BB localization. In particular, the ability to provide richer object hypotheses than 2D BB in the form of viewpoint estimates is typically connected to significantly sacrificing 2D localization performance in comparison to state-of-the-art object detectors.

In this paper, we aim to combine the best of both worlds, namely, to leverage performance from one of the most powerful 2D object class detectors to date,

and a 3D object class representation that allows for fine-grained 3D object and scene reasoning. In this way, we hope to benefit from the compact and rich 3D representation while retaining the robustness in matching to real-world images.

Our paper makes several contributions. First, we propose a 3D version of the powerful Deformable Part Model, combining the representational power of 3D modelling with robust matching to real-world images. Second, we demonstrate that our model delivers richer object hypotheses than 2D BB, in the form of viewpoint estimates of arbitrary granularity, and part localization consistent across viewpoints, outperforming prior work. Third, in contrast to previous work in multi-view recognition, we show competitive performance to state-of-the-art techniques for 2D BB localization.

1.1 Related Work

The recognition of object classes from multiple viewpoints while estimating their pose has received increasing attention, due to its potential to aid scene-level geometric reasoning [24]. It mainly comes in two different flavors. The first flavor models 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, and varies from shape templates [21] over implicit shape models [14] and HOG templates [25] to constellation [19] and deformable part models [26, 23, 27]. Neighboring views are either treated independently [25, 19], connected by means of feature tracking [14], or considered jointly in a convex optimization framework [26, 23, 21, 27]. While view-based multi-view approaches achieve remarkable results in predicting a discrete set of object poses [26, 23, 21, 27], they typically require evaluating a large number of view-based detectors at test time for good performance, resulting in considerable runtime complexity (e.g., 32 shape templates [21] or 36 constellation models [19]). In contrast, our $3D^2PM$ is able to synthesize appearance models for viewpoints of arbitrary granularity on the fly, resulting in significant speed-ups (see Sect. 3.5).

The second flavor embraces the 3D nature of object classes, maintaining an explicit representation of the 3D placement of individual features [15, 16, 28, 29, 22] or object parts [17, 18, 20]. 3D geometry is either provided in the form of a depth sensor [16, 28, 29], structure-from-motion [22] or 3D CAD models [18, 20] during training, and modelled either non-parametrically [16, 28, 29, 22] or in the form of 3D Gaussian distributions [18, 20]. While these 3D representations constitute more compact and more faithful descriptions of object classes than their view-based counterparts, they typically can not compete with modern object class detectors optimized for 2D bounding box localization such as the DPM [1].

In this paper, we aim to overcome this limitation of 3D object class representations, by designing a 3D version of the deformable part model [1]. To that end, we reformulate the DPM as a structured output prediction problem [27], and represent part appearance as well as spatial deformations in 3D. As a consequence, our formulation models part deformations as true 3D distributions, and allows to synthesize part appearance models for arbitrary viewpoints.

2 3D Deformable Part Models

This section introduces our model, a part based model that is a conditional distribution over 3D parts. It can be seen as a 3D extension of the DPM [1]. Although the standard DPM has proven successful for object detection, it is ignorant to 3D object geometry. By encoding the underlying 3D object structure we obtain a compact model with a smaller number of parameters. At the same time we hope to obtain a model that is more descriptive of the 3D object itself.

We describe our model successively. In Sec. 2.1 we explain the conditional random field (CRF) model and fix notation. Sec. 2.2 describes the pairwise terms of the CRF that are drawing on the 3D part displacement model. Sec. 2.3 introduces three versions of the unary term. We propose a discrete model and two continuous ones. Model learning is introduced in Sec. 2.4 and inference in Sec. 2.5. The final model allows for arbitrary fine viewpoint estimation because of the 3D part displacement and continuous appearance model. In the experiments we carefully analyze the contributions of the individual modeling components.

Probably the most related model is described in [27]. While their model also uses a structured SVM objective to jointly optimize for detection and viewpoint estimation and includes 3D constraints across viewpoints, the resulting model still is a collection of 2D DPM models. As a result [27] can infer the viewpoint for a discrete set that has to be specified already during training time. The model described here however results in a full 3D model with a continuous appearance model, allowing for arbitrarily fine viewpoint estimation at test-time. Also, the number of parameters to be learned in [27] is far larger than for our model.

2.1 Preliminaries

Training data contains tuples $\{I_i, y_i\}_{i=1,\dots,N}$, with I the image and y the output variable consisting of three parts $y = (y^l, y^v, y^b) \in \mathcal{Y}$: y^b specifying the bounding box given by its upper, lower, left and right boundary; $y^v \in [0, 360)$ denoting the viewing angle; $y^l \in \{-1, 1, 2, \dots, C\}$ denoting the class membership of the object (-1 in case no object of interest is present). We use a star shaped conditional random field (CRF) to model the dependency of the output variable y on image evidence I. We have a collection of $M + 1$ parts, $p_0, p_1, \dots, p_M$, where $p_0 = (x_0, y_0, z_0)$ denotes the observed root part and the remaining parts are latent. Their values $p_i = (x_i, y_i, z_i)$ are defined relative to the root 3D position p_0. Thus, the conditional distribution for a given image I and model parameters β and the latent space denoted by $h = (p_1, \dots, p_M)$ has the following form:

$$p(y, h \mid I, \beta) = Z(I, \beta)^{-1} \exp\left(- \sum_{i=0}^{M} \langle \beta_i^u, \psi^u(I, y, p_i) \rangle - \sum_{i=1}^{M} \langle \beta_i^p, \psi^p(I, p_0, p_i) \rangle \right) \quad (1)$$

where $\beta_i^u \in \mathbb{R}^D, i = 0, \dots, M$ are parameters of the unary term, $\beta_i^p \in \mathbb{R}^{D_p}, i = 1, \dots, M$ are parameters of pairwise terms, and the ψ^u, ψ^p are the unary and pairwise feature functions, that we specify in the next two sections. For a more compact notation we use β for stacked unary and pairwise terms.

2.2 Three-dimensional displacement model

The pairwise terms in equation (1) are related to the displacement of part p_i w.r.t so called anchor part positions $v_i = (v_{ix}, v_{iy}, v_{iz})$, which are defined w.r.t. the root part p_0. These displacements are defined as 3D Gaussians with diagonal covariance matrix Σ_i:

$$p(p_i|p_0, v_i) \propto \exp\left(-\frac{1}{2}\left((x_i, y_i, z_i)^\top - (v_i + \mu_i)\right)^\top \Sigma_i^{-1}\left((x_i, y_i, z_i)^\top - (v_i + \mu_i)\right)\right) \tag{2}$$

While the anchors are fixed during initialization they allow for free movement of the parts in 3D. To obtain the pairwise terms ψ^p the 3D part displacement distribution is projected onto a particular viewpoint (see Fig. 1). As a general perspective projection can result in a non-Gaussian distribution we restrict ourselves to a scaled orthographic projection Q^v instead. While this is clearly an approximation it works well in practice in particular when the object is relatively far way. To get more accurate approximation, we introduce a separate scaled orthographic projection Q_i^v for each part p_i. Each Q_i^v has a unique scaling factor related to the depth of the anchor of p_i in this particular view. For a given part p_i the projected 2D part displacement distribution has a mean $\mu_i^v = Q_i^v \mu_i$ and covariance $\Sigma_i^v = (Q_i^v)\Sigma_i(Q_i^v)^T$. As Q_i^v is a linear transformation, the resulting 2D part displacement distribution remains a Gaussian distribution. This distribution is fully parameterized with 6 parameters per part. This is in contrast e.g. to [27] where separate displacement models are trained for K different viewpoints resulting in $4K$ parameters. Thus this model is compact but also comes with fewer degrees of freedom.

Going from 3D to 2D, each part $p_i = (x_i, y_i, z_i)$ in 2D is parameterized as $p_i^v = (u_i, v_i, l_i)$, where $(u_i, v_i) = Q_i^v p_i$ and l_i is the resolution in image space and is typically fixed for each part to be at twice the resolution of the root filter [1]. The root itself $p_0 = (u_0, v_0, l_0)$ has a (u_0, v_0) position in the 2D image and a resolution parameter l_0. Recall the output variables y^l, y^v, y^b. There is a deterministic relationship between the position and resolution of the root filter p_0 and the bounding box of the training example, which is why we will speak of p_0 and y^b interchangeably. Finally, the pairwise term is defined as follows, closely following the original DPM formulation [1]:

$$\langle \beta_i^p, \psi^p(I, p_0, p_i)\rangle = \left\langle \left((\Sigma_i^v)_{11}^{-1}, (\mu_i^v)_1, (\Sigma_i^v)_{22}^{-1}, (\mu_i^v)_2, (\Sigma_i^v)_{12}^{-1}\right), \left(-du^2, -du, -dv^2, -dv, -2dudv\right)\right\rangle \tag{3}$$

where (du, dv) are the offsets of the projected part from the projected anchor, measured in the image plane.

2.3 Appearance model

Although our goal is to have a full 3D object model with continuous appearance representation, we start by introducing a discrete appearance model and then describe two continuous versions. We divide the viewing circle $[0, 360)$ into K different bins. Conceptually the easiest model is to train K different filters, one for each bin and then use this filter as a unary factor for all those y where y^v falls into the bin. We are going to refer to this model as $3D^2$PM-D, where "D" stands for discrete appearance bins. Thus the factor ψ_k^u for an appearance bin

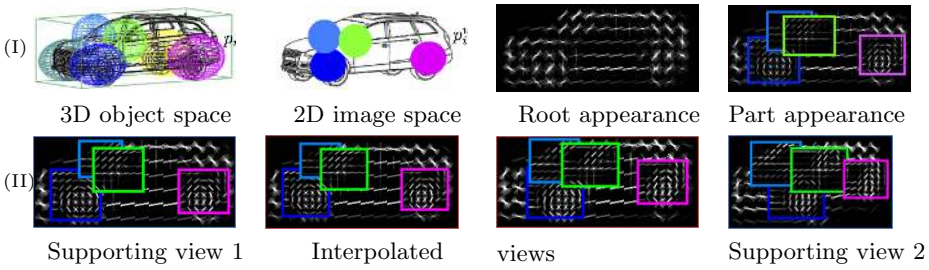

Fig. 1. Part displacement distributions and continuous appearance model. (I) Left to right: Learned 3D part displacement distributions, part projections in an arbitrary view (some 3D parts not visible due to occlusion), root and part appearances at the given view. (II) Continuous appearance model. First and last column: two supporting views, middle: two interpolated views.

k is represented through the root factor $\psi^u_{0,k}$ and the part factors $\psi^u_{i,k}$ for that particular bin. In this paper we use HOG [10] as features. The unary terms for the different parts are given by:

$$\langle \beta^u_i, \psi^u(I, y, p_i) \rangle = \sum_{k=1}^{K} \mathbf{1}_{y^v \in k} \langle \beta^u_{i,k}, \psi_{i,k}(I, y, h) \rangle. \tag{4}$$

In order to arrive at a continuous viewpoint model we need to specify unary potentials for arbitrary viewpoint y^v beyond the K bins. For this we interpolate among the unary filters of the appearance bins (called 3D^{2}PM-C in the following, see Fig. 1). The continuous models allow for establishing arbitrarily fine viewpoint estimation as the appearance is not restricted to a set of K bins. Note that in this case there are no actual bins as we do not perform binning but rather use the appearance in so called supporting viewpoints among which the continuous appearance model interpolates. Only for naming consistency with 3D^{2}PM-D, we will refer to the supporting viewpoints as bins. We explore two interpolation schemes, namely linear interpolation and exponential interpolation. In the linear interpolation scheme, the continuous appearance is defined as a linear combination of the appearance in the discrete appearance bins

$$\langle \beta^u_i, \psi^u(I, y, p_i) \rangle = \sum_{k=1}^{K} \alpha_k \langle \beta^u_{i,k}, \psi_{i,k}(I, y, h) \rangle \tag{5}$$

where $\alpha_k \propto \angle(y^v, y^v_k)$ is proportional to the angular distance between the viewpoint y^v of the example and the viewpoint of the k-th appearance bin y^v_k. In the exponential interpolation scheme we assign exponential weighting to the unaries of the appearance bins.

$$\langle \beta^u_i, \psi^u(I, y, p_i) \rangle = \sum_{k=1}^{K} e^{-d^2(y^v, y^v_k)} \langle \beta^u_{i,k}, \psi_{i,k}(I, y, h) \rangle \tag{6}$$

where $d(y^v, y^v_k) \propto \angle(y^v, y^v_k)$. In the experiments described below we analyze and compare all three models in terms of detection performance and viewpoint estimation accuracy.

2.4 Model learning

We train our models using a latent variable structured SVM objective with margin-rescaling [30]

$$\min_{\beta^u,\beta^p\geq 0,\xi\geq 0} \frac{1}{2}\|\beta\|^2 + \frac{C}{N}\sum_{n=1}^{N}\xi_n \tag{7}$$

$$\text{sb.t.} \quad \max_{h_n}\langle\beta,\psi(I_n,y_n,h_n)\rangle - \max_{\bar{h}}\langle\beta,\psi(I_n,\bar{y},\bar{h})\rangle \geq \Delta(y_n,\bar{y}) - \xi_n, \forall\bar{y}\in\mathcal{Y}. \tag{8}$$

The loss function Δ is task dependent and as we are interested in object localization and accurate viewpoint estimation, in this work we use a linear combination of an object detection error and a term measuring viewpoint estimation accuracy

$$\Delta(y,\bar{y}) = \alpha\Delta_{VOC}(y,\bar{y}) + (1-\alpha)\Delta_{vp}(y,\bar{y}). \tag{9}$$

Here Δ_{VOC} is the object detection error. We use the same form of Δ_{VOC} as in [31, 27] which penalizes detections with insufficient ground truth overlap. The viewpoint estimation loss Δ_{vp} can attain two forms, depending if we are interested into performing viewpoint classification or angular viewpoint estimation. In the case of viewpoint classification, we use 0/1 loss and in the case of angular viewpoint estimation we use angular precise loss $\Delta_{vp}(y,\bar{y}) = \frac{\angle(y^v,\bar{y}^v)}{180^\circ}$. 3D^{2}PM-D uses the 0/1 viewpoint loss and thus optimizes for viewpoint classification and 3D^{2}PM-C uses the continuous viewpoint loss and thus optimizes for angular precise viewpoint estimation. We use $\alpha = 0.5$ in the experiments.

The learning leverages on 3D information from CAD data. Following [19] we use wireframe-like non-photorealistic rendered images of CAD models.

CAD data is used for unsupervised part initialization as well. We perform the same part initialization as DPM but in 3D. First, a 3D grid of possible part placements is defined. The parts have predefined size (sx, sy, sz) and each candidate part location gets an appearance energy score, which is a sum of the appearances of the corresponding projected parts across views. From this set, the top k parts, which attain the highest appearance score are chosen.

Training. We employ stochastic gradient descend to solve the non convex max-margin optimization problem. We use an expectation-maximization like technique where we iterate between inferring the latent variables h, while the model β is fixed and training the model β for fixed latent variables (the non-root parts).

2.5 Inference

Finally, test time inference is the following problem $\text{argmax}_{y,h}\langle\beta,\psi(I,y,h)\rangle$ and can be computed using the max-product algorithm. This will infer the position of an object, its viewpoint and also all parts. The inference problems for 3D^{2}PM-D and 3D^{2}PM-C differ. In the case of 3D^{2}PM-D, every test example gets assigned to one of the K appearance bins (viewpoint classes) also determining it's viewpoint label. As in [1, 27] parts are inferred using efficient distance transform. The maximum scoring bin determines the viewpoint of the test example.

In the case of 3D^{2}PM-C viewpoint inference is a continuous problem due to the continuous nature of the appearance model. In practice we resort to establishing inference on an arbitrarily fine viewpoint resolution (obtained by interpolation) as enabled by the continuous nature of the appearance model. Note that the decision at which viewpoint resolution the model is evaluated is done during test-time and not before training the model. In the experiments we will report on the accuracy for viewpoint estimation depending on the number of appearance bins used during training.

3 Experimental Evaluation

In this section we thoroughly evaluate our model, by successively adding 3D information, going beyond plain 2D bounding box localization. While gradually going towards full 3D object model, we first consider the task of coarse viewpoint estimation, where we compare 3D^{2}PM-D and 3D^{2}PM-C models (Sect. 3.1). In a second step, and different from previous work in multi-view recognition, we aim at providing arbitrarily fine viewpoint estimation in real world images by leveraging on the full 3D nature of our 3D^{2}PM-C model (Sect. 3.2 and 3.3). While improving state-of-the-art results on standard benchmarks for fine viewpoint estimation, we give a detailed analysis of different aspects of 3D^{2}PM-C and describe an coarse to fine viewpoint estimation inference (Sect. 3.5).

Even-though the focus is on viewpoint estimation, we realize that a viewpoint estimation system has to be performant on the task of object detection as well, and hence provide object detection results for all viewpoint benchmarks as well as on the challenging Pascal VOC 2007 [32] dataset, where we show superior detection performance to all previous 3D object models in the literature.

We further compare different data sources for training, namely, real world images and synthetic images in the form of rendered CAD models, motivated by previous work leveraging CAD models [18, 19, 27]. We show that synthetic data can improve viewpoint estimation due its ability to provide perfect annotation despite its appearance statistics that differs from real images. We use 41 commercial cars (www.doschdesign.com) and 43 bicycles (www.sketchup.google.com).

3.1 Coarse-grained viewpoint estimation

We start by evaluating the 3D^{2}PM-D on coarse viewpoint estimation, phrased as a multi-class classification problem (viewpoint binning). We report results for cars and bicycles of 3D Object classes [15], a challenging benchmark data set tailored towards multi-view recognition (8 different viewpoint bins). In all experiments, we train from real images provided by the respective dataset as well as CAD data, which serve as a 3D proxy for our model and provide natural 3D constraints across different views of the same instance. We follow the testing protocols of [15, 32, 23] and report Mean Precision of Pose Estimation (MPPE) [23] as a measure of viewpoint classification accuracy (diagonal average of confusion matrix). We evaluate detection performance using Average Precision (AP) as established in the Pascal VOC [32] challenge.

AP/MPPE	2D Models				3D Models			
	[21]	[23]	[27]^1	[27]^2	[18]	[20]	[22]	3D^{2}PM-D
cars	- / 86.1	96.0 / 89.0	99.7 / 96.3	99.9 / 97.9	76.7 / 70.0	90.4 / 84.0	99.2 / 85.3	99.6 / 95.8
bicycles	- / 80.8	91.0 / 90.0	95.0 / 96.4	97.6 / 98.9	69.8 / 75.5	- / -	- / -	94.1 / 96.0

Table 1. Viewpoint estimation (in MPPE [23]) and object detection (in AP) results on car and bicycle class from 3D Object classes [15] dataset. [27]^1 and [27]^2 refer to DPM-3D-Constraints and DPM-VOC+VP respectively.

Results. Tab. 1 shows results, including 3D^{2}PM-D (last col.), recent successful 3D object models (Col. 5-7), and various 2D models (Col. 1-4), notably the state-of-the-art 2D object detector DPM-VOC+VP [27] (Col. 4). We observe 3D^{2}PM-D to achieve 95.8% and 96.0% MPPE on cars and bicycles, respectively, outperforming all previous work using 3D object models (85.3% cars [22] and 75.5% bicycles [18]). Comparing to 2D models 3D^{2}PM-D performs on par with DPM-3D-Constraints(96.3% and 96.4%, Col. 3) and it is slightly worse than DPM-VOC+VP (97.9%, 98.9%). The object detection results show similar tendency. 3D^{2}PM-D with 99.6% AP and 94.1% AP on cars and bicycles outperforms all previous work using 3D models (99.2% and 69.8%), it performs on par with DPM-3D-Constraints (99.7% and 95.0%), and is slightly worse than DPM-VOC+VP (99.9% with 97.6%).

We stress that 3D^{2}PM-D is trained with full 3D part displacement model across appearance bins. It shows remarkable viewpoint estimation and 2D detection performance on this dataset in comparison to the DPM-VOC+VP model, which is a 2D model and directly optimizes for the given task. On the other hand, even though DPM-3D-Constraints is a more complex model than 3D^{2}PM-D, as it models part displacement independently in each view, it shows comparable performance with our 3D^{2}PM-D model.

Summary. In conclusion, 3D^{2}PM-D outperforms previous 3D models and achieves competitive performance to the state-of-the-art multi-view 2D object detectors [27, 23], despite being less complex due to its full 3D representation.

3.2 Fine-grained viewpoint estimation

In a next round of experiments, we go one step further and evaluate 3D^{2}PM-D and 3D^{2}PM-C w.r.t fine-grained viewpoint estimation. To this end, we use EPFL Multi-view cars [25] due to the more fine-grained annotations. The data set contains 20 sequences of cars imaged from a full circle of 360 degrees. Angular viewpoint annotations are approximate. We follow [25] and use the first 10 sequences for training and test on the other 10. Viewpoint estimation is again phrased as multiclass classification, but we now vary the granularity of viewpoint sampling. Thus now we have models with k bins for $k \in \{8, 12, 16, 18, 36\}$. In each model, the bin centers have equi-distant spacing of $\frac{360}{k}$. As the annotations are continuous, we evaluate the 3D^{2}PM-C models with linear (3D^{2}PM-C Lin) and exponential (3D^{2}PM-C Exp) appearance interpolation as well.

Results. Table 2 compares object detection and viewpoint classification performance of our 3D^{2}PM-D and 3D^{2}PM-C models with linear and exponential interpolation to previously published results. For viewpoint estimation, 3D^{2}PM-D with 8 appearance bins achieves 78.5% MPPE which is 5% better than the

AP/MPPE	Ozuysal et al [25]	Lopez et al [23]	3D^{2}PM-D	3D^{2}PM-C Lin	3D^{2}PM-C Exp
8 bins	- / -	91.0 / 73.7	99.4 / 78.5	97.8 / 78.3	98.1 / 77.9
12 bins	- / -	- / -	97.9 / 75.5	98.3 / 76.2	98.4 / 77.3
16 bins	85.0 / 41.6	97.0 / 66.0	99.0 / 69.8	97.5 / 69.0	98.0 / 69.1
18 bins	- / -	- / -	99.2 / 71.8	99.3 / 71.2	99.2 / 70.5
36 bins	- / -	- / -	99.3 / 45.8	99.2 / 52.1	99.5 / 53.5

Table 2. Detection (AP) and viewpoint estimation (MPPE [23]) (EPFL dataset).

state-of-the-art result 73.7% MPPE of [23]. 3D^{2}PM-C Lin and 3D^{2}PM-C Exp achieve comparable accuracy of 78.3% and 77.9%, respectively, also improving over previous work. 3D^{2}PM-D with 16 bins achieves 69.8% MPPE which is by 4% better than the previous state-of-the-art result of [23] and by 28.2% better than [25]. 3D^{2}PM-C Lin and 3D^{2}PM-C Exp with MPPE of 69% and 69.1%, respectively, similarly outperform previous work and are on par with 3D^{2}PM-D. In terms of detection, 3D^{2}PM-C Lin and 3D^{2}PM-C Exp with 8 bins achieve 97.8% AP and 98.1% AP, outperforming the result 91.0% of [23], while 3D^{2}PM-D achieves 99.4% AP which is in the range of the 3D^{2}PM-C models. For 16 bins, 3D^{2}PM-D, 3D^{2}PM-C Lin and 3D^{2}PM-C Exp achieve 99%, 97.5% and 98% AP and collectively outperform the state-of-the-art results 97% of [23] and 85% of [25]. Increasing viewpoint granularity from 8 to 36 appearance bins, we observe that detection performance stays roughly the same in the range of 98-99% AP for all 3D^{2}PM variants. For viewpoint classification, performance seems to drop. This is in fact an artifact of the MPPE measure accounting only for 0/1 error. Interestingly, the 3D^{2}PM-C Lin and 3D^{2}PM-C Exp with 36 appearance bins achieve MPPE of 52.1% and 53.5% and outperform the 45.8% AP of 3D^{2}PM-D confirming that the continuous appearance model can be more suited for fine viewpoint estimation as it accounts for fine appearance variations.

As EPFL Multi-view cars offer angular viewpoint annotations, we also evaluate the Median Angular Error (MAE) as in [22], quantifying the more meaningful continuous angular error rather than 0/1 error as it is the case for MPPE.

Tab. 3 reports MAE for our 3D^{2}PM models comparing to state-of-the-art. Since the 3D^{2}PM-C uses continuous appearance models, we evaluate it at a finer viewpoint sampling of k bins. This enables us to explore the advantage of having continuous appearance modelling in comparison to discrete 2D modelling employed by the rest of the models and all previous work. Our 3D^{2}PM-D, 3D^{2}PM-C Lin and 3D^{2}PM-C Exp models with 8 bins achieve 12.9°, 11.1° and 9.6° MAE outperforming by almost 15° the best published result of 24.8° of [22].

Analyzing the different granularities of viewpoint estimation, MAE reduces as we go from coarse (8 bins) to finer viewpoint sampling (36 bins) and the 3D^{2}PM-

MAE	Glasner et al [22]	3D^{2}PM-D	3D^{2}PM-C Lin	3D^{2}PM-C Exp
8 bins	24.8	12.9	11.1	9.6
12 bins		9.0	7.8	8.8
16 bins		7.2	6.9	7.5
18 bins		6.2	5.6	6.9
36 bins		5.8	4.7	4.7

Table 3. Fine viewpoint estimation in MAE [22] (EPFL dataset).

Fig. 2. Object detection and 3D pose estimation. Example car and bicycle detections on Pascal 2007 [32]. Learned part distributions. The 3D part detections are color coded.

C models achieve 4.7° MAE with 36 bins which is better than 5.8° of 3D^{2}PM-D, again confirming the intuition that modelling objects in their natural form (in this case the continuous viewpoint appearance) leads to improved performance.

3.3 Arbitrarily fine viewpoint estimation

We proceed to evaluate the ability of the 3D^{2}PM-C to generate viewpoint estimates of arbitrary fine granularity, enabled by its continuous appearance representation. We use EPFL Multi-view cars as the only dataset providing angular accurate viewpoint annotation. Our goal is to understand better the 3D^{2}PM-C models and analyze its behavior in different settings. We train 3D^{2}PM-C with k bins, where again $k \in \{8, 12, 16, 18, 36\}$ and try to interpolate from the starting k viewpoints to arbitrarily fine viewpoint resolution. We go dense on the viewpoint sampling as the dataset permits, i.e. up to the label noise of the dataset. We evaluate at viewpoint resolution of 5°, 8°, 10°, 20°, 22.5°, 30°, and 45°.

Results. Fig. 3 and Tab. 4 give the results for 3D^{2}PM-C Lin (left) 3D^{2}PM-C Exp (right). At a coarse level, it is evident that for both models better viewpoint estimation is obtained at finer viewpoint resolution regardless of k. Exploring the other dimension in the plot (number of appearance bins), going from 8 to 36 bins increases performance.

Considering the respective best results, the 3D^{2}PM-C Lin and 3D^{2}PM-C Exp with 36 bins provide 4.7° MAE, better than any other result reported in the literature approaching the dataset label noise. The 3D^{2}PM-C models with 8 model viewpoints achieve remarkably good viewpoint estimation performance (MAE = 9.6° of 3D^{2}PM-C Exp and 11.1° of 3D^{2}PM-C Lin) despite large angular distance between the bins, even on the finest evaluation level of 5° resolution.

Comparing linear vs. exponential appearance interpolation, both models achieve comparable performance on the finer viewpoint resolution levels (5°,

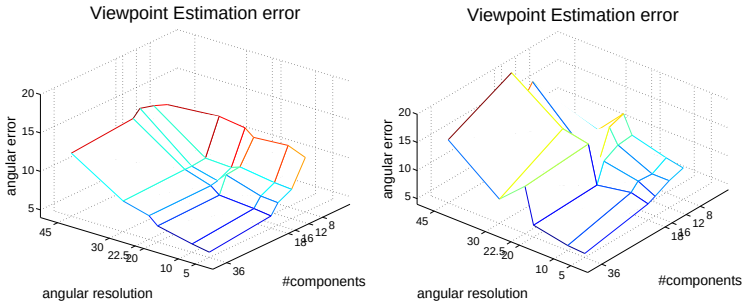

Fig. 3. Graphical representation of viewpoint classification results, left - linear interpolation, right - exponential. The number of components is the number bins.

AP/MAE	at 5°	at 10°	at 20°	at 22.5°	at 30°	at 45°
3D^{2}PM-C Lin 8	96.8 / 11.1	95.6 / 12.0	96.1 / 11.7	96.9 / 12.6	97.3 / 13.1	97.8 / 12.6
3D^{2}PM-C Lin 12	98.5 / 7.8	98.6 / 8.0	95.9 / 8.7	97.9 / 8.3	98.3 / 8.5	97.2 / 13.3
3D^{2}PM-C Lin 16	97.8 / 6.9	98.2 / 7.1	96.8 / 8.5	97.5 / 7.4	97.6 / 8.3	95.0 / 13.8
3D^{2}PM-C Lin 18	99.1 / 5.6	99.0 / 5.9	99.3 / 6.3	98.6 / 7.2	98.5 / 8.3	97.0 / 12.9
3D^{2}PM-C Lin 36	99.2 / 4.7	98.8 / 5.1	98.5 / 6.1	98.2 / 7.1	98.0 / 8.0	97.5 / 12.2

Table 4. Fine-grained viewpoint estimation in MAE [26] (EPFL dataset).

10°, 20°, 22.5°). However, for coarser viewpoint resolution and wider viewpoint spacing among the bins, exponential interpolation provides worse results than linear interpolation.

Summary . While the 3D^{2}PM-C model is a simple continuous model, it achieves good performance even when starting from wide angular spacing among the model viewpoint (appearance) bins.

3.4 CAD vs. real image data

We want to explore the performance impact of using CAD data, as they have unrealistic appearance but perfect viewpoint annotation. Thus we train models on synthetic data only (synthetic), on real & synthetic (mixed) and on real data where we use CAD data only for model initialization. We do experiments with 3D^{2}PM-D and 3D^{2}PM-C models with 8, 18 and 36 bins on the EPFL dataset.

Tab. 5 gives the result. Synthetic models with 36 bins achieve very good viewpoint classification performance of 5.9° and 7.9° for 3D^{2}PM-C and 3D^{2}PM-D, respectively, while also achieving good detection results of 96.3% and 95.2% AP. Adding real data (mixed) leads to improved results of 5.8° MAE for 3D^{2}PM-D and 5.1° MAE for 3D^{2}PM-C, while real data only with 6.4° MAE and 5.6° performs worse, speaking in favor of using CAD data with accurate annotation.

3.5 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). As a result, inference becomes slow. Thus we propose a

AP / MAE	3D^{2}PM-D			3D^{2}PM-C		
	mixed	real	synthetic	mixed	real	synthetic
8 bins	99.4 / 12.9	99.2 / 13.1	95.4 / 14.2	97.8 / 12.6	98.3 / 13.7	94.5 / 13.9
18 bins	99.2 / 6.2	99.1 / 7.4	94.9 / 9.8	99.3 / 6.3	98.2 / 7.0	95.6 / 7.1
36 bins	99.3 / 5.8	99.0 / 6.4	95.2 / 7.9	98.8 / 5.1	98.7 / 5.6	96.3 / 5.9

Table 5. Real vs. mixed data setting on 3D^{2}PM-C.

AP / MAE	at 5°	#atomic operations
3D^{2}PM-C b36 full	99.2 / 4.7	2.20×10^{10}
3D^{2}PM-C b36 coarse to fine	99.0 / 7.0	0.48×10^{10}
3D^{2}PM b12	97.6 / 7.5	2.20×10^{10}
3D^{2}PM b18	98.0 / 6.9	2.20×10^{10}

Table 6. Detection (AP) and vp. estimation (MAE). Full vs. coarse-to-fine inference.

speed up by using a coarse-to-fine inference that minimizes atomic operations while not sacrificing too much performance. We use a greedy, binary search-like scheme that recursively partitions the space of candidate viewpoints considered.

Results. Tab. 6 gives results on EPFL. 3D^{2}PM-C with 36 bins and full inference (row 1) is compared to the same model with coarse-to-fine inference (row 2) starting at 12 viewpoints. The last two rows are 3D^{2}PM-C models trained with 12 and 18 bins, respectively, evaluated on 5°. While achieving almost 5 times faster runtime (0.48×10^{10} vs. 2.2×10^{10} atomic operations), we obtain comparable detection results to the full inference model (99.0% AP and 99.2% AP with coarse-to-fine and full inference) and slightly worse viewpoint estimation results (7.0° vs. 4.7°). Interestingly, compared to models trained with 12 and 18 appearance bins, we achieve better results while attaining much smaller number of atomic operations. More sophisticated methodologies for search space pruning, such as Branch and Rank [33], could further improve that trade-off.

3.6 Pascal VOC 2007 detection

While previous work on 3D Object models typically reports results on multi-view benchmarks, we evaluate detection performance of the 3D^{2}PM model on the standard detection benchmark Pascal VOC 2007 [32]. This is important, since viewpoint estimation is inherently dependent on accurate object localization. Some visual results are shown in Fig. 2.

3D^{2}PM-D achieves 61.2% AP on cars, outperforming the previous best 3D Object Model result of 32% AP of [22] by a large margin. Still, the 2D DPM-VOC model [27] achieves 4% better result of 65.7% AP. DPM-3D-Constraints [27] achieves 63.1%. On bicycles, DPM-VOC and DPM-3D-Constraints achieve 61.3% and 56.8% AP which is better than 3D^{2}PM-D's 52.1% AP. However, given its full 3D nature, 3D^{2}PM's performance is encouraging.

3.7 Ultra-wide baseline matching

Lastly, we leverage the 3D nature of our model and the resulting ability to match parts across different viewpoints. We quantify this ability in the form of the ultra-wide baseline matching task established by [20].

Azimuth	SIFT	Zia [20]	DPM-3D Const. 12	DPM-3D Const. 20	3D^{2}PM-D 12	3D^{2}PM-D 20
45 °	2.0%	55.0%	49.1%	54.7%	47.2%	58.5%
90 °	0.0%	60.0%	42.9%	51.4%	54.3%	77.1%
135 °	0.0%	52.0%	55.2%	51.7%	44.8%	58.6%
180 °	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%

Table 7. Ultra-wide baseline matching performance, measured by the fraction of correctly estimated fundamental matrices.

Tab. 7 gives results comparing to pure SIFT matches, [20], and [27]. 3D^{2}PM-D with 20 parts with 66% of correctly estimated matrices, provides better performance than the DPM-3D-Constraints with 20 parts(54%) and better than 50% of [20]. We observe a significant improvement of 17.3% to DPM-3D-Constraints 12 and 29.6% to [20] on the wide baseline matching task of 180°, which we attribute to the ability of 3D^{2}PM-D to better distinguish opposing views.

4 Conclusion

In this work we have built a 3D object model which combines features from the most powerful object detector to date, the DPM, and a 3D object class representation. Being the first extension of the DPM to a full 3D object model, the 3D^{2}PM leverages on 3D information provided by CAD data, performing viewpoint estimation at arbitrarily fine granularity and achieves state-of-the-art results on viewpoint estimation and wide-baseline matching tasks. At the same time, it performs on par with state-of-the-art 2D object detectors w.r.t. detection performance. Therefore, the 3D^{2}PM takes a step towards bridging the gap between object detection and higher level tasks like scene understanding and 3D object tracking.

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. Felzenszwalb, P.F., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. PAMI (2010)
2. Marr, D., Nishihara, H.: Representation and recognition of the spatial organization of three-dimensional shapes. Proc. Roy. Soc. London B **200** (1978) 269–194
3. Brooks, R.: Symbolic reasoning among 3-d models and 2-d images. Artificial Intelligence **17** (1981) 285–348
4. Pentland, A.: Perceptual organization and the representation of natural form. Artificial Intelligence **28** (1986)
5. Lowe, D.: Three-dimensional object recognition from single two-dimensional images. Artificial Intelligence (1987)
6. Stark, L., Hoover, A., Goldgof, D., Bowyer, K.: Function-based recognition from incomplete knowledge of shape. In: WQV93. (1993)
7. Green, K., Eggert, D., Stark, L., Bowyer, K.: Generic recognition of articulated objects through reasoning about potential function. CVIU (1995)

8. Fergus, R., Perona, P., Zisserman, A.: Object class recognition by unsupervised scale-invariant learning. In: CVPR. (2003)
9. Leibe, B., Leonardis, A., Schiele, B.: Robust object detection with interleaved categorization and segmentation. IJCV **77** (2008) 259–289
10. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR. (2005)
11. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. IJCV (2008)
12. Ess, A., Leibe, B., Schindler, K., Gool., L.V.: Robust multi-person tracking from a mobile platform. PAMI (2009)
13. Wojek, C., Roth, S., Schindler, K., Schiele, B.: Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In: ECCV. (2010)
14. Thomas, A., Ferrari, V., Leibe, B., Tuytelaars, T., Schiele, B., Van Gool, L.: Towards multi-view object class detection. In: CVPR. (2006)
15. Savarese, S., Fei-Fei, L.: 3D generic object categorization, localization and pose estimation. In: ICCV. (2007)
16. Yan, P., Khan, S., Shah, M.: 3D model based object class detection in an arbitrary view. In: ICCV. (2007)
17. Su, H., Sun, M., Fei-Fei, L., Savarese, S.: Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In: ICCV. (2009)
18. Liebelt, J., Schmid, C.: Multi-view object class detection with a 3D geometric model. In: CVPR. (2010)
19. Stark, M., Goesele, M., Schiele, B.: Back to the future: Learning shape models from 3d cad data. In: BMVC. (2010)
20. Zia, Z., Stark, M., Schindler, K., Schiele, B.: Revisiting 3d geometric models for accurate object shape and pose. In: 3dRR-11. (2011)
21. Payet, N., Todorovic, S.: From contours to 3d object detection and pose estimation. In: ICCV. (2011)
22. Glasner, D., Galun, M., Alpert, S., Basri, R., Shakhnarovich, G.: Viewpoint-aware object detection and pose estimation. In: ICCV. (2011)
23. Lopez-Sastre, R.J., Tuytelaars, T., Savarese, S.: Dpm revisited: A performance evaluation for object category pose estimation. In: ICCV-WS CORP. (2011)
24. Bao, S.Y., Savarese, S.: Semantic structure from motion. In: CVPR. (2011)
25. Ozuysal, M., Lepetit, V., Fua, P.: Pose estimation for category specific multiview object localization. In: CVPR. (2009)
26. Gu, C., Ren, X.: Discriminative mixture-of-templates for viewpoint classification. In: ECCV. (2010)
27. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3d geometry to deformable part models. In: CVPR. (2012)
28. Arie-Nachimson, M., Basri, R.: Constructing implicit 3D shape models for pose estimation. In: ICCV. (2009)
29. Sun, M., Xu, B., Bradski, G., Savarese, S.: Depth-encoded hough voting for joint object detection and shape recovery. In: ECCV. (2010)
30. Yu, C.N.J., Joachims, T.: Learning structural svms with latent variables. In: International Conference on Machine Learning (ICML). (2009)
31. Blaschko, M., Lampert, C.: Learning to localize objects with structured output regression. In: ECCV. (2008)
32. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL VOC 2007 Results (2007)
33. Lehmann, A., Gehler, P., Van Gool, L.: Branch&rank: Non-linear object detection. In: BMVC. (2011)