3D Object Class Detection in the Wild

Bojan Pepik[1] Michael Stark[1] Peter Gehler[2] Tobias Ritschel[1] Bernt Schiele[1]

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

Abstract

Object class detection has been a synonym for 2D bounding box localization for the longest time, fueled by the success of powerful statistical learning techniques, combined with robust image representations. Only recently, there has been a growing interest in revisiting the promise of computer vision from the early days: to precisely delineate the contents of a visual scene, object by object, in 3D. In this paper, we draw from recent advances in object detection and 2D-3D object lifting in order to design an object class detector that is particularly tailored towards 3D object class detection. Our 3D object class detection method consists of several stages gradually enriching the object detection output with object viewpoint, keypoints and 3D shape estimates. Following careful design, in each stage it constantly improves the performance and achieves state-of-the-art performance in simultaneous 2D bounding box and viewpoint estimation on the challenging Pascal3D+ [50] dataset.

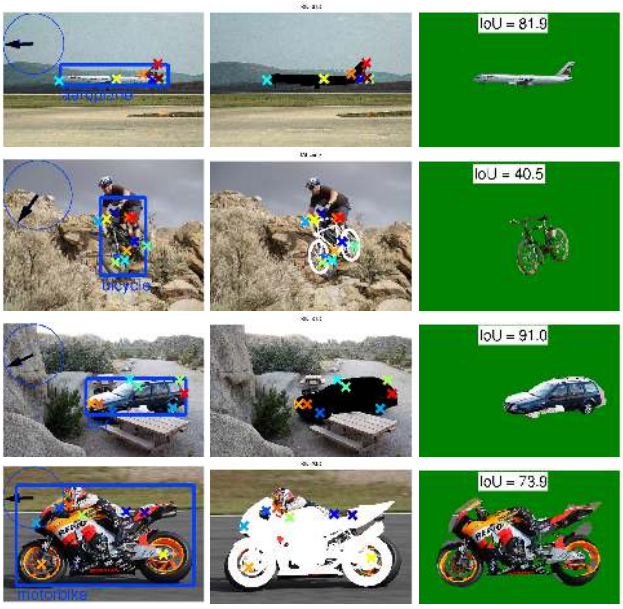

Figure 1. Output of our 3D object class detection method. (Left) BB, keypoints and viewpoint estimates, (center) aligned 3D CAD prototype, (right) segmentation mask.

1. Introduction

Estimating the precise 3D shape and pose of objects in a scene from just a single image has been a long standing goal of computer vision since its early days [33, 8, 39, 32]. It has been argued that higher-level tasks, such as scene understanding or object tracking, can benefit from detailed, 3D object hypotheses [12, 49, 19] that allow to explicitly reason about occlusion [41, 57, 6] or establish correspondences across multiple frames [52]. As a consequence, there has been an increasing interest in designing object class detectors that predict more information than just 2D bounding boxes, ranging from additional viewpoint estimates [44, 22, 31, 50] over 3D parts that correspond across viewpoints [42, 47] to the precise 3D shape of the object instance observed in a test image [56, 55, 35].

So far, these efforts have lead to two main results. First, it has been shown that simultaneous 2D bounding box localization and viewpoint estimation, often in the form of classification into angular bins, are feasible for rigid object classes [47, 43, 28, 37, 2, 27]. These *multi-view object class detectors* typically use view-based [31, 42] or coarse 3D geometric [51, 40, 18, 35, 34] object class representations that are designed to generalize across variations in object shape and appearance. While these representations have shown remarkable performance through the use of joint training with structured losses [42, 40], they are still limited with respect to the provided geometric detail.

Second, and more recently, it has been shown that highly detailed 3D shape hypotheses can be obtained by *aligning 3D CAD model instances* to an image [56, 30, 3, 29]. These approaches are based on a large database of 3D CAD models that ideally spans the entire space of object instances expected at recognition time. Unfortunately, the added detail comes at a cost: first, these approaches are targeted only towards specific object classes like cars and bicycles [56], chairs [3], or pieces of IKEA furniture [30, 29], limiting their generality. Second, they are typically evaluated on

datasets with limited clutter and occlusion [56], such as 3D Object Classes [43], EPFL Multi-View Cars [37], or particular subsets of PASCAL VOC [13] without truncation, occlusion, or "difficult" objects [3].

In this work, we aim at joining the two directions, multi-view detection and 3D instance alignment, into *3D object class detection in the wild* – predicting the precise 3D shape and pose of objects of various classes in challenging real world images. We achieve this by combining a robust, part-based object class representation based on RCNNs [20] with a small collection of 3D prototype models, which we align to the observed image at recognition time. The link between a 2D image and a 3D prototype model is established by means of 2D-3D keypoint correspondences, and facilitated by a pose regression step that precedes rigid keypoint alignment.

As a result, the presented method predicts the precise 3D shape and pose of all PASCAL3D+ [50] classes (Fig. 1), at no loss in performance with respect to 2D bounding box localization: our method improves over the previous best results on this dataset [15] by 21.2% in average precision (AP) while simultaneously improving 12.5% in AAVP (Sect. 4.4) in joint object localization and viewpoint estimation [42]. In addition, projecting the 3D object hypotheses provided by our system onto the image plane result in segmentation masks that are competitive with native segmentation approaches, highlighting the accuracy of our 3D shape estimates.

This paper makes the following contributions. First, to our knowledge, we present the first method for 3D object class detection in the wild, achieving precise 3D shape and pose estimation at no loss of 2D bounding box localization accuracy compared to state-of-the-art RCNN detectors. Second, we design a four-stage detection pipeline that is explicitly tailored towards 3D object class detection, based on a succession of *(i)* robust 2D object class detection, *(ii)* continuous viewpoint regression, *(iii)* object keypoint detection and *(iv)* 3D lifting through rigid keypoint alignment. Third, we give an in-depth experimental study that validates the design choices at each stage of our system. Crucially, and in contrast to previous work [42, 40], we demonstrate that enriching the output of the object detector does not incur any performance loss: the final 3D detections yield the same AP as stage (i) and improved AAVP over stage (ii), even though significant geometric detail is added. And fourth, we demonstrate superior performance compared to state-of-the-art in 2D bounding box localization, simultaneous viewpoint estimation, and segmentation based on 3D prototype alignment, on all classes of the PASCAL3D+ dataset [50].

2. Related work

Our approach draws inspiration from four different lines of work, each of which we review briefly now.

2D Object class detection. Recently, RCNNs (regions with convolutional neural network features) have shown impressive performance in image classification and 2D BB localization [20], outperforming the previous de-facto standard, the deformable part model (DPM) [15], by a large margin. Our pipeline is hence built upon an RCNN-based detector that provides a solid foundation to further stages. To our knowledge, our model is the first to extend a RCNN-based object class detector towards 3D detection.

Multi-view object class detection. In recent years, computer vision has seen significant progress in multi-view object class detection. Successful approaches are mostly extensions of proven 2D detectors, such as the implicit shape model [47, 53, 28, 46], the constellation model [45, 44], an the deformable part model [22, 31, 42, 40, 51, 18, 35, 34], resulting in both view-based [44, 22, 31, 42] and integrated, 3D representations [38, 40, 51, 18, 35, 56, 55, 34, 4] that reflect the 3D nature of object classes.

Our work follows a different route, and decomposes the 3D detection problem into a sequence of simple, but specialized pipeline stages, each optimized for performance. From the multi-view detection literature, we take inspiration mostly from continuous viewpoint regression [37, 22, 40, 56, 4], which we use as the second stage of our pipeline. In contrast to previous work, however, our pipeline does not end with a viewpoint estimate, but rather uses it to guide the next stage, 3D lifting. As we show in our experiments, the last stage benefits from the intermediate viewpoint regression, and even improves the regressor's estimate.

3D Instance alignment. Methods that align 3D CAD model instances to a test image [56, 30, 3, 29] are receiving increasing attention, due to their ability to yield highly precise estimates of 3D object shape and pose (sometimes referred to as fine pose estimation [30, 29]). These methods are based on a large number of 3D CAD model instances that are rendered from a large set of viewpoints, in order to sufficiently cover appearance variations. While the resulting complexity can be alleviated by selecting discriminative exemplar patches [3] or sharing of 3D parts [29], it is still linear in the cross-product of instances and viewpoints, limiting scalability. In contrast, we focus on capturing only the major modes of shape variation in the form of a hand-full of prototypical 3D CAD models per object class. In addition, our representation is based on only a small number of 3D keypoints (on average 10 per object class) that are not only shared among instances, but also matched to image evidence in a viewpoint-invariant way [57]. As a result, we can increase the accuracy of our 3D lifting stage by adding more CAD models, without retraining our pipeline.

Keypoint-based methods. The concept of deriving 3D information from predicted 2D keypoints is well known in human body pose estimation [1, 7], and has also

been successfully applied to estimating the rough pose of birds [14, 5] or fitting deformable 3D shape models [56, 57]. Our work draws from this idea in order to find a rigid alignment of a prototypical 3D CAD model to an image.

3. 3D Object class detection

In this section, we describe our 3D object class detection pipeline. Given a single test image as an input, it can not only predict the 2D bounding box (BB) of each object in the image, but also yields estimates of their 3D poses as well as their 3D shape, represented relative to a set of prototypical 3D CAD models. Fig. 1 gives example results. A schematic overview of our method is shown in Fig. 2.

The following subsections provide a walk-through of our pipeline. We start with robust 2D object class detection (Sect. 3.1). We then add viewpoint information (Sect. 3.2). Next, we localize a set of 3D object keypoints in the 2D image plane (Sect. 3.3) that provides the basis for our last stage: 3D lifting (Sect. 3.4). It combines all estimates of the previous stages into a final, 3D object class detection result. Since this last step depends crucially on the quality of the intermediate stages, we highlight the important design choices that have to be made in each subsection.

3.1. 2D Object class detection

RCNNs [25, 20] have shown remarkable performance in image classification and 2D BB localization, leading to state-of-the-art results on the Pascal VOC [13] and ImageNet [11] datasets. As precise BB detection and 2D alignment are crucial requirements for being able to infer 3D geometry, we adopt RCNNs as the first stage of our pipeline.

Specifically, we use the implementation of Girshick et al. [20] (RCNN). It consists of three steps: generation of BB proposals, feature extraction using the intermediate layers of a CNN, and subsequent training of a one-vs-all SVMs.

The selective search method [48] provides several object candidate regions $o \in \mathcal{O}$ in an image. These are passed into a CNN [25] and its unit activations in separate layers are extracted as feature representation for each region. The RCNN [20] uses the responses of either the last convolutional (conv5) or one of the two fully connected layers (fc6, fc7). A linear SVM is trained for every object class, with the positive examples being the regions with a certain intersection-over-union (IoU) overlap R with the ground truth and the negative examples the regions with IoU ≤ 0.3 with the ground truth. At test time, the RCNN provides for each image I a set of object detections $o = [o^b, o^c, o^s]$, where o^b is the BB, o^c the object class, and o^s the score.

Empirical results in [20] on the Pascal VOC 2007 and 2010 datasets identify fc7 features and $R = 1$ as the best set of parameters. We compared the combination of intermediate feature responses and values of R on the Pascal3D+ [50] dataset and found the same setting to perform best.

3.2. Viewpoint estimation

An essential cue for performing the transition from 2D to 3D is an accurate estimate of the 3D pose of the object, or, equivalently, of the viewpoint under which it is imaged. We represent the viewpoint of an object $o^v \in [0, 360)$ in terms of azimuth angle a. Several approaches can be taken to obtain a viewpoint estimate, treating it either as a discrete or continuous quantity. We discuss the discrete version first, mainly to be comparable with recent work. However we argue that due to the continuous nature of the viewpoint the problem should be treated as a continuous regression problem. As the experiments will show (Sect. 4.2), this treatment outperforms the discrete variants allowing for a much finer resolution of the viewpoint estimate.

Discrete viewpoint prediction. A large body of previous work and datasets on multi-view object class detection [43, 21, 42, 50] use a discretization of the viewpoint into a discrete set of V classes, typically focusing on a single angle (azimuth). The task is then to classify an object hypothesis into one of the $v \in \{1, \ldots, V\}$ classes. While this defeats the continuous nature of the problem, it has the benefit of giving a reduction to a multi-class classification problem for which efficient methods exist.

We conjecture that a CNN representation will be discriminative also for viewpoint estimation and explore two different CNN variants to test this hypothesis. First, we use the pre-trained CNN from Section 3.1 and replace the last linear SVM layer for object detection with one for viewpoint estimation. Discretizing the viewpoints in V classes results in V different classifiers for every object category. During test time, we choose the class with the maximum score. We refer to this method as RCNN-MV. We explore a second variant (CNN-MV), a multi-view CNN trained end-to-end to jointly predict category and viewpoint. The CNN parameters are initialized from a network trained on ImageNet [11] for object category classification and is then trained using logistic loss and backpropagation [24].

Continuous viewpoint prediction. While discrete viewpoint prediction is the de-facto standard today, we believe that angular accurate viewpoint estimation is both more natural and leads to better performance, which is confirmed by the empirical results in Sect. 4.2.

We again use the intermediate layer responses of a CNN, pretrained for detection (Section 3.1), as the feature representation for this task. From these features, we regress the azimuth angle directly. More formally, let us denote with ϕ_i the features provided by a CNN on region o_i depicting an object of category c. Let o^a represent the azimuth of the region and w^a the azimuth regressor for class c. We use a least squares objective

$$w^a = \underset{w}{\operatorname{argmin}} \, ||o_i^a - \phi_i^\top w||_2^2 + \lambda ||w||_p^2, \qquad (1)$$

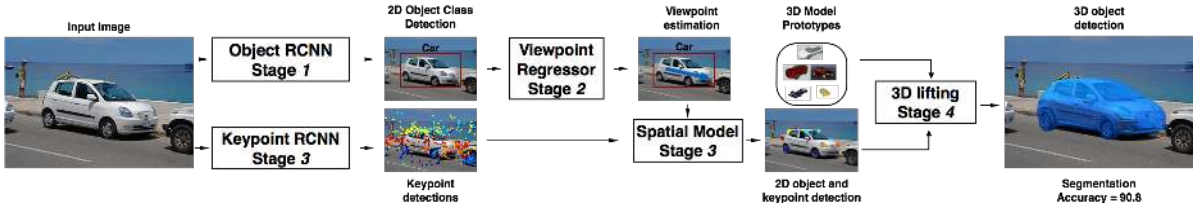

Figure 2. Our 3D object class detection pipeline.

and test three different regularizers: ridge regression ($p = 2$), lasso ($p = 1$), and elastic net. We refer to the regressors as RCNN-Ridge, RCNN-Lasso and RCNN-ElNet. In our experiments, we found that these are the best performing methods, confirming that the CNN features are informative for viewpoint estimation, and that the continuous nature of the problem should be modeled directly.

3.3. Object keypoint detection

While an estimate of the 3D object pose in terms of azimuth angle (Sect. 3.2) already conveys significant geometric information beyond a 2D BB, it is not enough to precisely delineate a 3D prototype model, which is the desired final output of our 3D object class detection pipeline. In order to ultimately do the lifting to 3D (Sect.3.4), our model relies on additional geometric information in the form of object keypoints. They establish precise correspondences between 3D object coordinates and the 2D image plane.

To that end, we design a set of object class specific keypoint detectors that can accurately localize keypoints in the 2D image plane. In connection with a spatial model spanning multiple keypoints, these detectors can deliver reliable anchor points for 2D-3D lifting.

Keypoints proposal and detection. Recently, it has been shown that powerful part detectors can be obtained by training full-blown object class detectors for parts [10]. Inspired by these findings, we once more turn to the RCNN as the most powerful object class detector to date, but train it for keypoint detection rather than entire objects. Since keypoints have quite different characteristics in terms of image support and feature statistics, we have to perform the following adjustments to make this work.

First, we find that the standard RCNN mechanism for obtaining candidate regions, selective search [48], is suboptimal for our purpose (Sect. 4.3), since it provides only limited recall for object keypoints. This is not surprising, since it has been designed to reliably propose regions for entire objects: it starts from a super-pixel segmentation of the test image, which tends to undersegment parts in most cases [23]. We hence propose an alternative way of generating candidate regions, by training a separate DPM [15] detector for each keypoint. To generate positive training examples we need to define a BB around each keypoint. We use a squared region centered at the keypoint that covers 30% of the relative size of the object BB. At test time, we can then choose an appropriate number of candidate keypoint regions by thresholding the DPM's dense sliding window detections.

Second, we find that fine-tuning the CNN on task-specific training data makes a difference for keypoint detection (Sect. 4.3). We compare two variants of RCNN keypoint detectors, both scoring DPM keypoint proposal regions using a linear SVM on top of CNN features. The first variant re-uses the CNN features trained for 2D object class detection (Sect. 3.1). The second one fine-tunes the CNN on keypoint data prior to feature computation.

Spatial model. Flexible part-based models are among the most successful approaches for object class recognition in numerous incarnations [17, 16, 15], since they constrain part positions to overall plausible configurations while at the same time being able to adapt to intra-class shape variation – both are crucial traits for the 3D lifting stage of our pipeline. Here, we start from the spatial model suggested by [3] in the context of localizing mid-level exemplar patches, and extend it for 3D instance alignment. This results in a simple, effective, and computationally efficient spatial model relating object with keypoint detections.

We define a spatial model that relates the position of keypoints to the position of the object center in the 2D image plane, resulting in a star-shaped dependency structure as in previous work [26, 15]. Specifically, for every different keypoint class p we estimate on the training data the average relative position around the object center o. Around this estimated mean position we define a rectangular region $N(o, p)$ of size proportional to the standard deviation of the relative keypoint positions in the training set. At test time, for a given object center o, for every part p we perform max-pooling in $N(o, p)$. This prunes out all keypoint detections outside of $N(o, p)$ and only retains the strongest one inside.

As the visibility and relative locations of keypoints changes drastically with object viewpoint, we introduce a number of viewpoint-specific components of this spatial model. During training, these components are obtained by clustering the viewpoints into C clusters, and estimating the mean relative keypoint position on each component.

At test time we resort to two strategies to decide on which component to use. We either use the viewpoint estimation (Sect.3.2) as a guidance for which one to use, or we

use the one with the best 3D detection objective (Sect. 3.4). Indeed, the guided version performs better (Sect. 4.3).

3.4. 3D Object class detection

The result of the previous stages is a combination of a 2D object BB (Sect. 3.1) plus a set of 2D keypoint locations (Sect. 3.3) specific to the object class. Optionally, the keypoint locations are also specific to viewpoint, by virtue of the viewpoint estimation (Sect. 3.2) and the corresponding spatial model component. This input can now be used to lift the 2D object class detection to 3D, resulting in a precise estimate of 3D object shape and pose.

We choose a non-parametric representation of 3D object shape, based on prototypical 3D CAD models for the object class of interest. Assuming known correspondences between keypoints defined on the surface of a particular model and 2D image locations, we can estimate the parameters of the projective transformation that gives rise to the image.

3D Lifting. We adopt the camera model from [50] and use a pinhole camera P always facing the center of the world, assuming the object is located there. Assuming a fixed field of view, the camera model consists of 3D rotation (pose) and 3D translation parameters. We parameterize the 3D pose as $o^v \in [0, 360) \times [-90, +90) \times [-180, 180)$, in terms of azimuth angle a, elevation angle e and the in-plane rotation θ. These three continuous parameters, fully specify the pose of a rigid object. The 3D translation parameters consist of the distance of the object to the camera D and the in-plane translation t.

The 3D lifting procedure jointly estimates the camera and the 3D shape of the object. Let us denote with $\{k^i\}$ the set of 2D keypoint predictions. Let $\{K_j^i\}$ be the corresponding 3D keypoints on the CAD model j and $\tilde{k}_j^i = PK_j^i$ denote the image projection of K_j^i. Then the CAD prototype c^* and camera P^* are obtained by solving

$$c^*, P^* = \underset{c,P}{\operatorname{argmin}} \sum_i^L ||k^i - \tilde{k}_c^i||. \quad (2)$$

We perform exhaustive search over the set of CAD models and solve for P using an interior point solver as in [50].

Initialization. The object viewpoint estimate is used to initialize the azimuth. The elevation is initialized using the category mean. We initialize $\theta = 0$. For the in-plane translation and 3D distance parameters, we solve Eq. 2 optimizing only for these parameters. This gives a good coarse initialization of the distance and the in-plane translation that is used later for the joint optimization of all parameters.

4. Experiments

In this section, we give an in-depth experimental study of the performance of our 3D object class detection pipeline, highlighting three distinct aspects. First, we validate the design choices at each stage of our pipeline, 2D object class detection (Sect. 4.1), continuous viewpoint regression (Sect. 4.2), keypoint detection (Sect. 4.3) and 3D lifting (Sect. 4.4), ensuring that each stage delivers optimal performance when considered in isolation. Second, we verify that adding geometric detail through adding more pipeline stages does not come at the cost of losing any performance, as it is often observed in previous work [27, 42, 40, 56]. And third, we compare the performance of our method to the previous state-of-the-art art, demonstrating significant performance gains in 2D BB localization, simultaneous localization and viewpoint estimation, and segmentation based on 3D prototype alignment. In contrast to previous work [56, 30, 3, 29], we evaluate the performance of our method for a variety of classes on challenging, real-world images of PASCAL VOC [13, 50].

Dataset. We focus our evaluation on the recently proposed Pascal3D+ [50] dataset. It enriches PASCAL VOC 2012 [13] with 3D annotations in the form of aligned 3D CAD models. The dataset provides aligned CAD models for 11 rigid classes (*aeroplane*, *bicycle*, *boat*, *bus*, *car*, *chair*, *dining table*, *motorbike*, *sofa*, *train*, and *tv monitor*) of the *train* and *val* subsets of PASCAL VOC 2012. The alignments are obtained through human supervision, by first selecting the visually most similar CAD model for each instance, and specifying the correspondences between a set of 3D CAD model keypoints and their image projections, which are used to compute the 3D pose of the instance in the image. Note that, while the 3D lifting stage of our pipeline (Sect. 3.4) is in fact inspired by this procedure, it is entirely automatic, and selects the best fitting 3D CAD model prototype without any human supervision. Throughout the evaluation, we use the *train* set for training and the *val* set for testing, as suggested by the Pascal3D+ [50].

State-of-the-art. We compare the performance of our pipeline to previous state-of-the-art results on the PASCAL3D+ dataset as reported in [50]. Specifically, we compare our results to two variants of the deformable part model (DPM [15]) that predict viewpoint estimates in the form of angular bins in addition to 2D BBs: *(i)* VDPM [50] trains dedicated mixture components for each angular viewpoint bin, using standard hinge-loss, and *(ii)* DPM-VOC+VP [42] optimizes mixture components jointly through a combined localization and viewpoint estimation loss [1]. This method has been shown to outperform previous work in multi-view detection by significant margins on 3D Object Classes [43] and PASCAL VOC 2007 cars and bicycles.

[1] The DPM-VOC+VP detections were provided by the authors of [42].

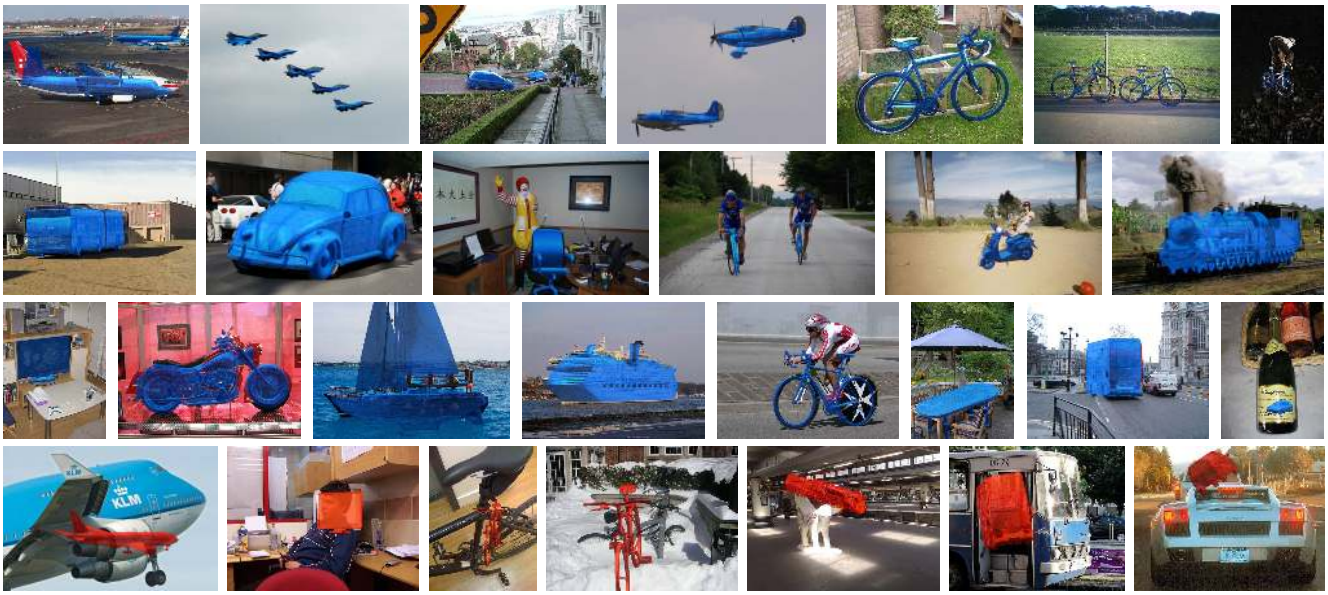

Figure 3. 3D CAD prototype alignment examples. (Blue) good alignments, (red) bad alignments. RCNN-Ridge-L fails mainly on truncated and occluded cases. For more 3D alignment visualizations please see the supplemental material.

4.1. 2D Bounding box localization

We start by evaluating the first stage of our pipeline, 2D object class detection (Sect. 3.1), in the classical 2D BB localization task, as defined by PASCAL VOC [13]. Fig 4 (left) compares the performance of our RCNN in its discrete multi-view variant RCNN-MV (cyan) to CNN-MV (green) and the state-of-the-art methods on this dataset, VDPM [50] (blue) and DPM-VOC+VP [42] (light blue). It reports the mean average precision (mAP) over all 11 classes of Pascal3D+ (per-class results are part of the supplemental material) for different numbers of discrete azimuth bins, as suggested by the PASCAL3D+ benchmark: VP_1, VP_4, VP_8, VP_{16} and VP_{24} denote the number of discrete viewpoint-dependent components of the respective model. Note that for the VP_1 case, the VDPM model reduces to the standard DPM [15] and RCNN-MV to the standard RCNN.

Results. We make the following observations. First, for VP_1, both RCNN (51.2%) and CNN (47.6%) outperform the previous state-of-the-art result of VDPM (29.6%) by significant margins of 21.6% and 18.0%, respectively, in line with prior reports concerning the superiority of CNN- over DPM-based detectors [20]. Second, we observe that the performance of VDPM and DPM-VOC+VP remains stable or even slightly increases when increasing the number of components (e.g., from 29.6% to 30.0% for VDPM and from 27.0% to 28.3% for DPM-VOC+VP and VP_{16}). Curiously, this tendency is essentially inverted for RCNN and CNN: performance drops dramatically from 51.2% to 30.8% and from 47.6% to 27.6% for AP_{24}, respectively.

Conclusion. We conclude that, while the training of per-viewpoint components is a viable strategy for DPM-based methods, RCNN-MV and CNN-MV both suffer from the decrease in training data available per component. We hence elect RCNN as the first stage of our 3D detection pipeline, leaving us with the need for another pipeline stage capable of estimating viewpoint.

4.2. Simultaneous 2D BB and viewpoint estimation

The original PASCAL3D+ work [50] suggests to quantify the performance of simultaneous 2D BB localization and viewpoint estimation via a combined measure, average viewpoint precision (AVP). It extends the traditional PASCAL VOC [13] detection criterion to only consider a detection a true positive if it satisfies both the IoU BB overlap criterion *and* correctly predicts the ground truth viewpoint bin ($\text{AVP} \leq \text{AP}$). This evaluation is repeated for different numbers of azimuth angle bins VP_4, VP_8, VP_{16} and VP_{24}. While this is a step in the right direction, we believe that viewpoint is inherently a continuous quantity that should be evaluated accordingly. We hence propose to consider the entire continuum of possible azimuth angle errors $D \in [0°, \ldots, 180°]$, and count a detection as a true positive if it satisfies the IoU and is within D degrees of the ground truth. We then plot a curve over D, and aggregate the result as the average AVP (AAVP). This measure has the advantage that it properly quantifies angular errors rather than equalizing all misclassified detections, and it alleviates the somewhat arbitrary choice of bin centers.

Fig. 4 (center) gives the results according to this measure, averaged over all 11 classes of PASCAL3D+

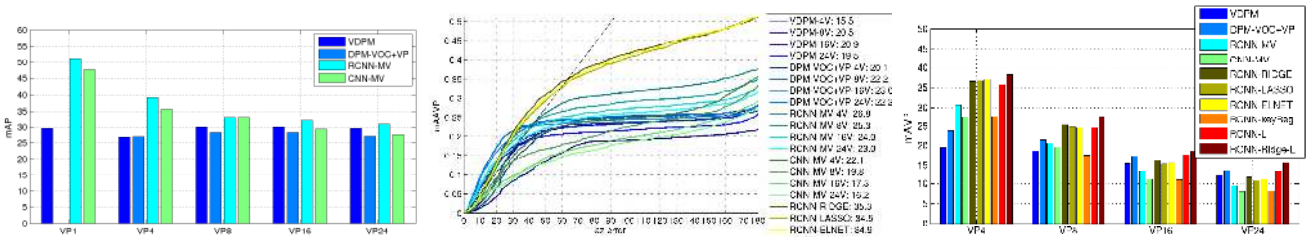

Figure 4. (Left) 2D BB localization on Pascal3D+ [50]. (Center, right) Simultaneous 2D BB localization and viewpoint estimation. (Center) continuous mAAVP performance, (right) discrete mAVP performance for VP_4, VP_8, VP_{16} and VP_{24}.

(per-class results are part of the supplemental material). Fig. 4 (right) gives the corresponding results in the original AVP measure for discrete azimuth angle binnings [50] as a reference. In both cases, we compare the performance of our different RCNN-viewpoint regressor combinations, RCNN-Ridge, RCNN-Lasso, and RCNN-ElNet, to the discrete multi-view RCNN-MV and CNN-MV, and the state-of-the-art methods VDPM and DPM-VOC+VP.

Results. We observe that in the mAAVP measure (Fig. 4 (left)), the RCNN-viewpoint regressor combinations outperform the previous state-of-the-art methods VDPM and DPM-VOC+VP by large margins. The best performing combination RCNN-Ridge (35.3%, brown) outperforms the best VDPM-16V (20.9%) by 14.4% and the best DPM-VOC+VP-16V (23.0%) by 12.3%, respectively.

The performance of VDPM and DPM-VOC+VP is stable or increasing for increasing numbers of components: VDPM-4V (15.5%) improves to VDPM-16V (20.9%), and DPM-VOC+VP-4 (20.1%) improves to DPM-VOC+VP-16V (23.0%). In contrast, performance decreases for RCNN-MV and CNN-MV: RCNN-MV-4V (26.8%) decreases to RCNN-MV-24V (23.0%), and CNN-MV-4V (22.1%) decreases to CNN-MV-24V (16.2%). Even though the best performing RCNN-MV-4V (26.8%) outperforms the previous state-of-the-art DPM-VOC+VP-16V (23.0%), it can not compete with the RCNN-viewpoint regressor combinations.

The same tendencies are also reflected in the original mAVP measure [50] (Fig. 4 (right)). While DPM-VOC+VP has a slight edge for the fine binnings (it outperforms RCNN-Ridge by 0.9% for VP_{16} and 1.9% for VP_{24}), RCNN-viewpoint regressor combinations dominate for the coarser binnings VP_4 and VP_8, followed by RCNN-MV, CNN-MV, VDPM, and DPM-VOC+VP.

Conclusion. The combination of RCNN and viewpoint regressor RCNN-Ridge provides a pronounced improvement in simultaneous 2D BB localization and viewpoint estimation compared to previous state-of-the-art (12.3% in mAAVP). Notably, it retains the original performance in 2D BB localization of the RCNN (51.2% in AP).

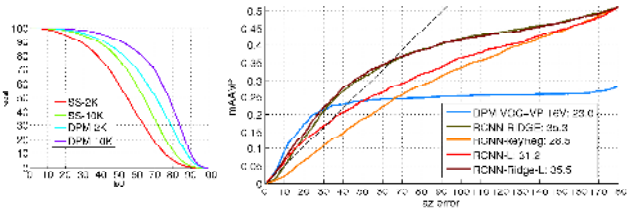

Figure 5. Left: 2D Keypoint region proposal quality. Right: Simultaneous 2D BB and viewpoint estimation with 3D lifting.

4.3. 2D Keypoint detection

We proceed by evaluating the basis for our 3D lifting stage, 2D keypoint detection (Sect. 3.3), in isolation. We use the keypoint annotations provided as part of Pascal3D+ [50], and train an RCNN keypoint detector for each of 117 types of keypoints distributed over 11 object categories. Since the keypoints are only characterized by their location (not extent), we evaluate localization performance in a way that is inspired by human body pose estimation [54]. For computing a precision-recall curve, we replace the standard BB IoU criterion for detection with an allowed distance P from the keypoint annotation, normalized to a reference object height H. We refer to this measure as Average Pixel Precision (APP). In all experiments, we use $H = 100$ and $P = 25$.

Region proposals. We first evaluate the keypoint region proposal method (Fig. 5 (left)), comparing selective search (SS) with the deformable part model (DPM [15]) at $K = 2000$ and $K = 10000$ top-scoring regions per image. The DPM is trained independently for each keypoint (for that purpose, we define the BB of each keypoint to be a square centered at the keypoint with area equal to 30% of the object area). Both DPM versions outperform the corresponding SS methods by large margin: at 70% IoU DPM with $K = 10000$ gives 30% more recall than SS-10K which is why we stick with these keypoint proposals for our 3D object class detection pipeline.

Part localization. Tab. 1 compares the performance of our RCNN keypoint detectors with the DPM keypoint proposal detectors alone, in APP. On average, the RCNN-FT keypoint detectors trained using the features from the

APP	aero plane	bike	boat	bus	car	chair	din. table	mot. bike	sofa	train	tv	AVG
DPM	19.2	36.2	8.9	26.4	14.3	3.1	4.0	24.2	7.6	8.5	6.1	14.4
RCNN	24.6	43.1	9.8	47.8	34.1	5.7	4.6	36.7	**14.3**	22.5	21.5	24.1
RCNN FT	**30.4**	**48.9**	**12.4**	**50.8**	**39.5**	**9.5**	**6.3**	**41.6**	14.0	**24.5**	**22.8**	**27.3**

Table 1. Part detection performance in APP.

sAcc	aero plane	bike	boat	bus	car	chair	din. table	mot. bike	sofa	train	tv	AVG
GT	58.3	32.0	57.9	84.9	79.6	53.5	63.1	69.3	64.7	70.5	80.7	65.0
RCNN-KeyReg	27.1	20.2	19.1	56.2	47.7	23.0	18.6	41.3	**46.4**	30.9	**70.0**	36.4
RCNN-L	30.3	22.0	**27.9**	60.5	44.2	24.9	24.4	46.3	41.9	37.5	45.6	36.9
RCNN-Ridge-L	**35.1**	**22.2**	26.9	**66.4**	**53.9**	**26.8**	**28.6**	**49.0**	44.8	**42.5**	58.7	**41.4**

Table 2. Segmentation accuracy on Pascal3D+.

sAcc	aero plane	bike	boat	bus	car	chair	din. table	mot. bike	sofa	train	tv	AVG
GT	40.3	27.9	36.2	75.0	59.3	34.9	16.0	59.0	25.2	57.0	72.5	45.7
O_2P [9]	48.2	32.5	29.6	**61.1**	46.7	12.4	12.4	**46.0**	17.0	36.7	41.6	34.9
O_2P+ [36]	**52.4**	**32.8**	**33.1**	60.5	**47.8**	12.8	**13.0**	44.5	16.7	**40.1**	40.7	**35.9**
RCNN-KeyReg	21.9	17.2	15.1	49.5	39.2	16.4	11.8	37.3	**21.9**	28.2	**60.9**	29.0
RCNN-L	26.7	18.8	17.5	53.9	36.7	16.2	6.4	43.5	16.3	35.5	49.7	29.2
RCNN-Ridge-L	27.7	20.1	19.9	59.0	41.7	**18.2**	7.8	44.4	18.5	37.9	51.1	31.5

Table 3. Segmentation accuracy on Pascal-context [36] dataset.

CNN fine-tuned on keypoint detection (27.3%) outperform the DPM (14.4%) by 12.9% APP providing a solid basis for our 3D lifting procedure.

4.4. 2D to 3D lifting

Finally, we evaluate the performance of our full 3D object class detection pipeline that predicts the precise 3D shape and pose. We first give results on simultaneous 2D BB localization and viewpoint estimation as before, but then move on to measuring the quality of our predicted 3D shape estimates, in the form of a segmentation task. We generate segmentation masks by simply projecting the predicted 3D shape (Fig. 1 (right)). We compare the performance of a direct 3D lifting (RCNN-L) of detected 2D keypoints with a viewpoint guided 3D lifting (RCNN-Ridge-L), and a baseline that regresses keypoint positions (RCNN-KeyReg) on top of an RCNN object detector rather than using keypoint detections.

Simultaneous 2D BB & VP estimation. Fig. 5 (right) compares the mAAVP performance of the lifting methods with the best viewpoint regressor RCNN-Ridge and the best previously published method DPM-VOC+VP-16V. Fig. 4 (right) gives the AVP_V [50] performance in comparison with all viewpoint classifiers and regressors.

RCNN-L (31.2% mAAVP) and RCNN-Ridge-L (35.5%) outperform both the RCNN-KeyReg (28.5%) and the DPM-VOC+VP-16V (23.0%) by considerable margins. RCNN-Ridge-L consistently outperforms RCNN-Ridge in terms of AVP_V (by 1.6%, 2.2%, 2.2%, and 4.1% for increasing V), thus improving over the previous pipeline stage. Furthermore, with 18.6% AVP_{16} and 15.8% AVP_{24} it also outperforms DPM-VOC+VP-16V (17.3%,13.6%, respectively), and achieving state-of-the-art simultaneous BB localization and viewpoint estimation results on Pascal3D+.

Segmentation. Tab. 2 reports the segmentation accuracy on Pascal3D+. We use the evaluation protocol of [50] with two differences. First, we evaluate inside the ground truth BB only to account for truncated and occluded objects. Second, we focus the evaluation on objects with actual ground truth 3D prototype alignment as that constitutes the relevant set of objects we want to compare on. Therefore, we report the performance of the ground truth aligned 3D CAD prototypes (GT) as well.

With 41.4% performance across all classes, RCNN-Ridge-L outperforms RCNN-L (36.9%) and the baseline RCNN-KeyReg (36.4%) by 4%, confirming the quality of the alignment. Fig. 3 illustrates successful 2D-3D alignments for different object classes, along with failure cases. Truncated and occluded objects represent a major part of the failures.

In Tab. 3 we go one step further and compare to native state-of-the-art segmentation methods (O_2P [9]), this time on the Pascal-context [36] dataset. We report the performance on the 11 classes from Pascal3D+ only. RCNN-Ridge-L with 31.5% is only slightly worse than O_2P+ (35.9%) although the latter is designed for segmentation.

Conclusion. We conclude that RCNN-Ridge-L achieves state-of-the-art simultaneous BB localization and viewpoint estimation performance on Pascal3D+ [50], outperforming the DPM-VOC+VP and the RCNN-Ridge regressor. It successfully predicts the 3D object shape which is confirmed by it's segmentation performance.

5. Conclusions

In this work we have build a 3D object class detector, capable of detecting objects of multiple object categories in the wild (Pascal3D+). It consists of four main stages: *(i)* object detection, *(ii)* viewpoint estimation, *(iii)* keypoint detection and *(iv)* 2D-3D lifting. Based on careful design choices, our 3D object class detector improves the performance in each stage, achieving state-of-the-art object BB localization and simultaneous BB localization and viewpoint estimation performance on the challenging Pascal3D+ dataset. At the same time, it predicts the 3D shape of the objects, as confirmed by it's segmentation quality. The final result is a rich 3D representation, consisting of 3D shape, 3D viewpoint, and 3D position automatically estimated using only 2D image evidence.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR'09*. 2

[2] M. Arie-Nachimson and R. Basri. Constructing implicit 3D shape models for pose estimation. In *ICCV'09*. 1

[3] 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 *CVPR*, 2014. 1, 2, 4, 5

[4] A. Bakry and A. Elgammal. Untangling object-view manifold for multiview recognition and pose estimation. In *ECCV*, 2014. 2

[5] P. based R-CNNs for Fine-grained Category Detection. Ning zhang, jeff donahue, ross girshick, trevor darrell. In *ECCV*, 2014. 3

[6] T. W. Bo Li and S.-C. Zhu. Integrating context and occlusion for car detection by hierarchical and-or model. In *ECCV*, 2014. 1

[7] L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 2

[8] R. A. Brooks. Symbolic reasoning among 3-d models and 2-d images. *AI'81*. 1

[9] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *ECCV'12*. 8

[10] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR'14*. 4

[11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR'09*. 3

[12] A. Ess, B. Leibe, K. Schindler, and L. V. Gool. Robust multi-person tracking from a mobile platform. *PAMI'09*. 1

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *IJCV'10*. 2, 3, 5, 6

[14] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV'11*. 3

[15] P. F. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI'10*. 2, 4, 5, 6, 7

[16] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV'05*. 4

[17] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR'03*. 4

[18] S. Fidler, S. Dickinson, and R. Urtasun. 3D object detection and viewpoint estimation with a deformable 3D cuboid model. In *NIPS'12*. 1, 2

[19] A. Geiger, M. Lauer, C. Wojek, C. Stiller, and R. Urtasun. 3d traffic scene understanding from movable platforms. *TPAMI'14*. 1

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR'14*. 2, 3, 6

[21] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV'11*. 3

[22] C. Gu and X. Ren. Discriminative mixture-of-templates for viewpoint classification. In *ECCV'10*. 1, 2

[23] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? In *BMVC'14*. 4

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014. 3

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS'12*. 3

[26] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1–3):259–289, 2008. 4

[27] J. Liebelt and C. Schmid. Multi-view object class detection with a 3D geometric model. In *CVPR'10*. 1, 5

[28] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3D feature maps. In *CVPR'08*. 1, 2

[29] J. J. Lim, A. Khosla, and A. Torralba. Fpm: Fine pose parts-based model with 3d cad models. In *ECCV*, 2014. 1, 2, 5

[30] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing IKEA Objects: Fine Pose Estimation. In *ICCV*, 2013. 1, 2, 5

[31] R. J. Lopez-Sastre, T. Tuytelaars, and S. Savarese. Deformable part models revisited: A performance evaluation for object category pose estimation. In *ICCV-WS CORP'11*. 1, 2

[32] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *AI'87*. 1

[33] D. Marr and H. K. Nishihara. Representation and recognition of the spatial organization of three-dimensional shapes. *PRSLB'78*. 1

[34] M.Hejrati and D.Ramanan. Analysis by synthesis: 3d object recognition by object reconstruction. In *CVPR'14*. 1, 2

[35] M.Hejrati and D.Ramanan. Analyzing 3D objects in cluttered images. In *NIPS'12*. 1, 2

[36] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR'14*. 8

[37] M. Ozuysal, V. Lepetit, and P. Fua. Pose estimation for category specific multiview object localization. In *CVPR'09*. 1, 2

[38] N. Payet and S. Todorovic. From contours to 3d object detection and pose estimation. In *ICCV'11*. 2

[39] A. P. Pentland. Perceptual organization and the representation of natural form. *AI'86*. 1

[40] B. Pepik, P. Gehler, M. Stark, and B. Schiele. 3DDPM - 3d deformable part models. In *ECCV'12*. 1, 2, 5

[41] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Occlusion patterns for object class detection. In *CVPR'13*. 1

[42] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Teaching 3d geometry to deformable part models. In *CVPR'12*. 1, 2, 3, 5, 6

[43] S. Savarese and L. Fei-Fei. 3D generic object categorization, localization and pose estimation. In *ICCV'07*. 1, 2, 3, 5

[44] M. Stark, M. Goesele, and B. Schiele. Back to the future: Learning shape models from 3D CAD data. In *BMVC'10*. 1, 2

[45] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV'09*. 2

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

[47] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *CVPR'06*. 1, 2

[48] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV'13*. 3, 4

[49] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3D scene modeling and inference: Understanding multi-object traffic scenes. In *ECCV'10*. 1

[50] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3D object detection in the wild. In *WACV'14*. 1, 2, 3, 5, 6, 7, 8

[51] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR'12*. 1, 2

[52] Y. Xiang, C. Song, R. Mottaghi, and S. Savarese. Monocular multiview object tracking with 3d aspect parts. In *European Conference on Computer Vision (ECCV)*, 2014. 1

[53] P. Yan, S. Khan, and M. Shah. 3D model based object class detection in an arbitrary view. In *ICCV'07*. 2

[54] Y. Yang and D. Ramanan. Articulated human detection with flexible mixtures of parts. *TPAMI'13*. 7

[55] E. Yoruk and R. Vidal. Efficient object localization and pose estimation with 3D wireframe models. In *3DRR'13*. 1, 2

[56] M. Z. Zia, M. Stark, B. Schiele, and K. Schindler. Detailed 3D representations for object recognition and modeling. *TPAMI'13*. 1, 2, 3, 5

[57] M. Z. Zia, M. Stark, and K. Schindler. Explicit occlusion modeling for 3d object class representations. In *CVPR'13*. 1, 2, 3