Data-Driven 3D Voxel Patterns for Object Category Recognition

Yu Xiang[1,2], Wongun Choi[3], Yuanqing Lin[3], and Silvio Savarese[1]
[1]Stanford University, [2]University of Michigan at Ann Arbor, [3]NEC Laboratories America, Inc.
yuxiang@umich.edu, {wongun, ylin}@nec-labs.com, ssilvio@stanford.edu

Abstract

Despite the great progress achieved in recognizing objects as 2D bounding boxes in images, it is still very challenging to detect occluded objects and estimate the 3D properties of multiple objects from a single image. In this paper, we propose a novel object representation, 3D Voxel Pattern (3DVP), that jointly encodes the key properties of objects including appearance, 3D shape, viewpoint, occlusion and truncation. We discover 3DVPs in a data-driven way, and train a bank of specialized detectors for a dictionary of 3DVPs. The 3DVP detectors are capable of detecting objects with specific visibility patterns and transferring the meta-data from the 3DVPs to the detected objects, such as 2D segmentation mask, 3D pose as well as occlusion or truncation boundaries. The transferred meta-data allows us to infer the occlusion relationship among objects, which in turn provides improved object recognition results. Experiments are conducted on the KITTI detection benchmark [17] and the outdoor-scene dataset [41]. We improve state-of-the-art results on car detection and pose estimation with notable margins (6% in difficult data of KITTI). We also verify the ability of our method in accurately segmenting objects from the background and localizing them in 3D.

1. Introduction

One of the major paradigms in modern object recognition consists of characterizing images with a list of 2D bounding boxes which correspond to the location and scale of the objects in the image. Recent methods have demonstrated that this task can be solved with a good degree of accuracy even when a large number of object categories is considered [10, 22, 18]. However, in many applications – autonomous driving is a notable example – recognizing objects as just 2D bounding boxes is not sufficient. In these applications, estimating the 3D object pose or figuring out the depth ordering of the objects from the observer is as important as (or even more important than) identifying the 2D locations of the objects. Moreover, in these scenarios, nuisances such as occlusions or truncation become dominant, and often one needs to recognize objects even when only a small portion of their surface is visible. The recently proposed KITTI benchmark [17] have been instrumental in highlighting the fact that object detection and 3D pose estimation tasks become extremely difficult when objects such as cars, bikes or trucks are to be recognized in the wild – that is within complex and cluttered urban scenes. Consider Fig. 1-top for instance, where cars occupy just a small portion of the image and most of them are heavily occluded by other cars. Except for a few exceptions [30, 24], most of the recent object detection methods have hard time in parsing out the correct configuration of objects from this kind of imagery.

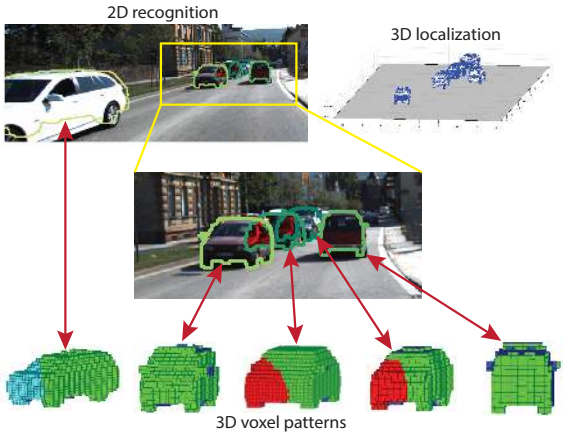


Figure 1. By introducing the 3D voxel patterns, our recognition framework is able to not only detect objects in images, but also segment the detected objects from the background, estimate the 3D poses and 3D shapes, localize them in the 3D space, and even infer the occlusion relationship among them. Green, red and cyan voxels are visible, occluded and truncated respectively.

In this paper, we present a novel recognition pipeline that addresses the key challenges above: i) it goes beyond 2D bounding box detection and is capable of estimating 3D properties of multiple detected objects such as 3D pose as well as their depth ordering from the observer; ii) it is designed to handle situations where objects are severely occluded by other objects or truncated because of a limited field of view; iii) it is capable of accurately estimating the

occlusion boundaries of each objects as well as inferring which portions of the object are occluded or truncated and which are not (see Fig. 1).

At the foundation of our recognition pipeline is the newly proposed concept of *3D Voxel Pattern* (3DVP). A 3DVP is a novel object representation that jointly captures key object properties which relates: i) appearance – the RGB luminance values of the object in the image; ii) 3D shape – the 3D geometry of the object expressed as a collection of 3D voxels; iii) occlusion masks – the portion of the object that is visible or occluded because of self-occlusions, mutual occlusions and truncations (Fig. 3(d)). Our approach follows the idea that luminance variability of the objects in the image due to intra-class changes and occlusions can be effectively modeled by learning a large dictionary of such 3DVPs whereby each 3DVP captures a specific shared "signature" of the three properties listed above (appearance, 3D shape and occlusions). Examples of 3DVPs in the dictionary are shown in Fig. 6. Inspired by a recent body of work [6, 4, 7, 27] that proposes to learn object detectors using clusters of 2D images that share similar appearance properties, in our recognition pipeline we train a bank of detectors using our dictionary of 3DVPs whereby each detector is trained from the appearance information associated to a specific 3DVP. Thus, these detectors are designed to localize objects in the image even when they are observed from arbitrary viewpoints or visible under severe occlusions. Moreover, because the 3DVPs retain shared properties about the object (specifically, 3D shape and occlusion masks), these can be transferred during the detection regime so as to recover the 2D segmentation mask of the object, its 3D pose as well as which portions of the objects are occluded and which are visible. Finally, and most critically, we use these properties to reason about object-object interactions and infer which object is an "occluder" and which is an "occludee". This in turn helps adjusting the confidence values of the detectors (e.g., if we know that an object is occluded and we predict which portion is occluded, this can help reinforce the presence of the occluder and its location; vice versa, the occluder can help support the presence of the occludee and the portion of the object that is occluded).

We believe our approach is particularly valuable in an autonomous driving scenario where vehicles' locations must be detected from images as well as vehicles' precise depth ordering and pose configurations must be inferred in 3D. For that purpose, we trained and tested our approach using the KITTI detection benchmark [17] – a large dataset of videos of cars driving in challenging urban scenes – and focused on recognizing cars and estimating their 3D properties. We also evaluated our method using the outdoor-scene dataset proposed in [41] – a dataset that has been specifically designed to test object detectors in presence of severe occlusions. We note that even if we only tested our method on the "car" category, our approach is general and can be extended to other rigid object categories. Our extensive experimental evaluation shows that: i) our approach based on 3D voxel patterns produces significant improvement over state-of-the-art results for car detection and 3D pose estimation on KITTI ($\sim 6\%$ for the hard test set); ii) our approach allows us to accurately segment object boundaries and infer which areas of the objects are occluded and which are not; we demonstrate that our segmentations results are superior than several baseline methods; iii) our approach allows us to localize objects in 3D and thus infer the depth ordering of the object from the camera's viewpoint.

2. Related Work

We review representative techniques in handling different challenges in object category recognition.

Shape variation. In order to handle the intra-class variability of shape, part-based object representations are introduced, such as the constellation model [12] and pictorial structures [11, 10]. Another direction is to discover and learn appearance models for object subcategories [6, 4, 7, 27], where object instances in a subcategory share similar visual appearance. In our recognition framework, we discover 3D voxel patterns, where object instances in a 3DVP share similar visibility pattern.

Viewpoint. Recent progresses in multiview object recognition can be roughly classified according to their ways of representing the object category. In 2.5D object representation, object parts or features are connected across views [33, 31, 32, 20]. While in 3D object representation, visual features are associated with explicit 3D models [42, 21, 25, 19, 13, 40]. The 3D models can either be built from a set of 2D images in different views [42, 19] or constructed using 3D CAD models [25, 40]. The new 3D object representation we introduce, i.e., 3D voxel pattern, utilizes 3D CAD models in the recognition pipeline.

Occlusion. In order to detect partially occluded objects, researchers have worked on training partial object detectors for visible parts of objects [38, 35, 15, 37, 41]. Since partial object detectors are not very robust, [38, 37, 41] also jointly reason about the presence of multiple objects in the scene. [43] and [30] explicitly consider the occluder when detecting the occluded object by introducing occlusion masks and occlusion patterns respectively. In all the previous works, only limited number of occlusion patterns are modeled. In contrast, we propose a data-driven approach to handle a large number of occlusion patterns.

Truncation. Objects can be truncated by image borders due to the limited field of view of the camera. Truncation is commonly handled by heuristics such as padding the image borders. An exception is [34], which detected truncated objects with a structured output regression. In our work, we handle truncation by leveraging our 3DVP representation

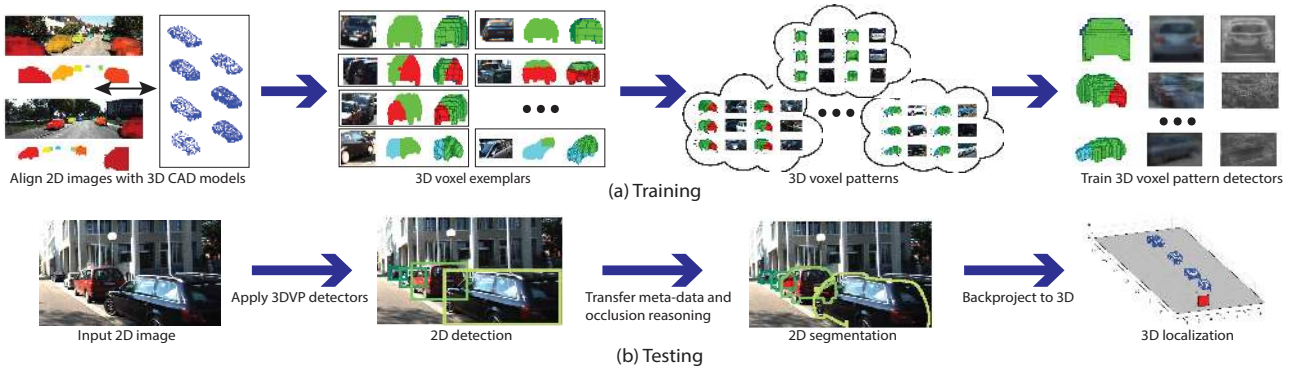


Figure 2. Overview of our object category recognition framework. (a) Training pipeline. (b) Testing pipeline.

which can be used to characterize truncated objects.

Nearest neighbor and deep neural network. Nearest neighbor based methods [26] and deep neural networks [22, 18] handle the above factors in object category recognition implicitly. Nearest neighbor is able to transfer meta-data of the training examples to testing objects, such as 2D segmentation mask, 3D shape, and so on. We inherit this advantage in our recognition framework. In deep neural networks, millions of parameters are learned from training data which has the ability to handle different aspects in object recognition without explicit modeling them. However, deep neural networks cannot estimate explicit 3D geometrical properties, such as 3D pose or occlusion boundaries.

3. Object Category Recognition with 3DVPs

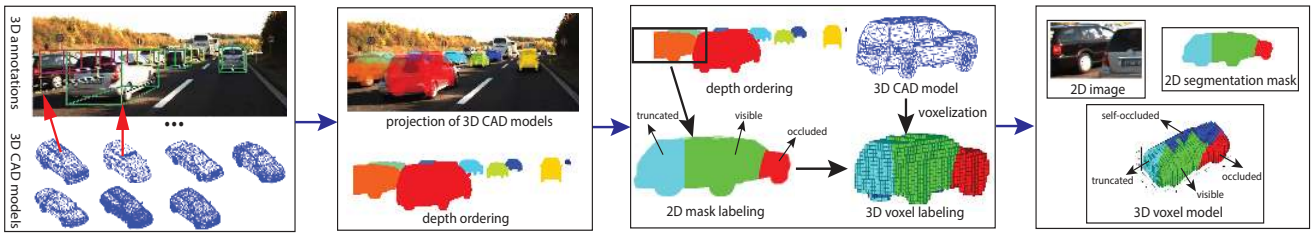
We propose a novel object recognition framework based on *3D Voxel Patterns* (3DVPs). 3DVPs are abstract 3D representations that capture patterns of visibility of an object category. The visibility of an object instance is represented by a *3D voxel exemplar*, which is a triplet of the 2D image of the object, its 2D segmentation mask and its 3D voxel model (see Fig. 4 for some examples).

In the training stage, we obtain 3D voxel exemplars in a data-driven approach (Sec. 3.1). Then we build a representative set of 3DVPs by clustering 3D voxel exemplars according to their visibility patterns (Sec. 3.2). Finally, we train a detector for each 3DVP (Sec. 3.3), which is specialized to detect objects with specific visibility patterns. Fig. 2(a) illustrates our training pipeline. Our approach is similar in spirit to [6, 4, 7, 27] that build subcategories based on 2D appearance patterns. Unlike these works, however, we learn detectors on the 3DVPs which capture explicit information about the visibility patterns of objects.

In the testing phase, after applying 3DVP detectors to an input image, we can transfer the meta-data associated with the 3DVPs, such as 2D segmentation mask, 3D pose or 3D shape, to the detected objects. These transferred meta-data enables us to perform different recognition tasks beyond 2D detection, such as object segmentation, pose estimation, 3D localization and occlusion reasoning. Fig. 2(b) illustrates our testing pipeline.

3.1. 3D Voxel Exemplars from Data

A 3D voxel exemplar captures the appearance, 3D shape and occlusion mask of an object. As long as a method can produce the 2D segmentation mask and the 3D voxel model of an object in the image, it can be used to build 3D voxel exemplars. For example, one could collect data with depth sensors or 3D scanners. However, it is difficult to scale to a large number of objects. Our solution is to utilize 3D CAD models in repositories on the web, such as the Trimble 3D Warehouse [2], and register these 3D CAD models with 2D images to build 3D voxel exemplars. In this way, we can obtain 3D voxel exemplars for tens of thousands of objects. We illustrate how to build 3D voxel exemplars for cars using the KITTI detection benchmark [17] in Fig. 3: 1) For each image in the training set, an object in the image is registered with a 3D CAD model selected from a pre-defined collection of models, where the model which has the closest aspect ratio with the ground truth 3D cuboid of the object instance is selected. The KITTI dataset [17] provides ground truth 3D annotations (cuboids) and camera parameters. Then we register the chosen 3D CAD model to the ground truth 3D cuboid associated to the object instance (Fig. 3(a)). 2) We project all the registered 3D CAD models onto the image plane using the camera parameters and obtain the depth ordering mask (Fig. 3(b)). 3) The depth ordering mask determines which pixel of the projected 3D CAD model is visible, occluded, or truncated. So we can generate a 2D segmentation mask for each object associated with visibility labels. We use green to color "visible" pixels, red to color "occluded" pixels, and cyan to color "truncated" pixels in the segmentation mask (Fig. 3(c)). To build the 3D voxel model for the object, we first voxelize the associated 3D CAD model. Then we check the status of each voxel in the voxelized 3D CAD model. From the camera viewpoint and the geometry of the 3D CAD model, we can figure out which voxels are visible or self-occluded (blue). For each visible voxel, we project it onto the depth ordering mask



(a) 3D CAD model association and registration (b) project 3D CAD models to the image (c) Label 2D segmenation mask and 3D voxel model (d) 3D voxel exemplar

Figure 3. Illustration of generating 3D voxel exemplars from images and annotations available from the KITTI detection benchmark [17].

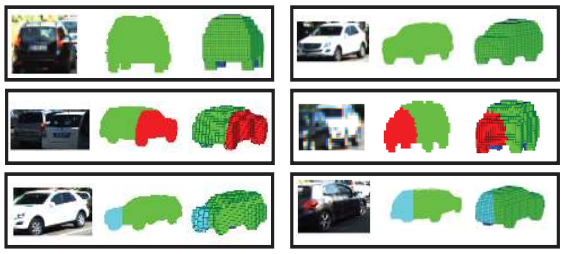


Figure 4. Examples of 3D voxel exemplars. Red indicates occlusion, and cyan indicates truncation.

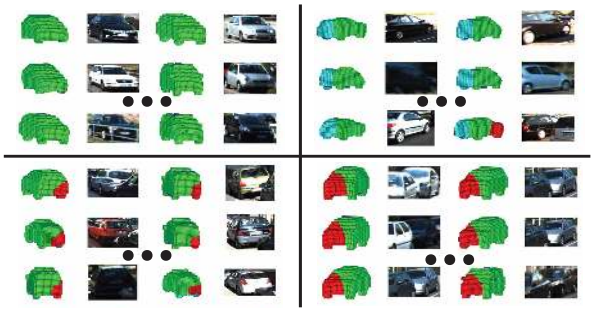


Figure 5. Examples of 3D clusters from the KITTI dataset.

to determine whether it is occluded or truncated (Fig. 3(c)). The result is a triplet called 3D voxel exemplar, which comprises the image of the object, the 2D segmentation mask of the object and the corresponding distribution of 3D voxels with associated visibility labels (Fig. 3(d)). More examples of the built 3D voxel exemplars are shown in Fig. 4. In our experiments, we use 7 3D car models from PASCAL3D+ [39] and obtain 57,224 3D voxel exemplars for car from the KITTI detection benchmark. We note that [3] also utilizes 3D CAD models with other cues to produce 2D segmentation masks for cars in KITTI ($\sim$ 1,000).

The 3D voxel representation has several good properties. First, by encoding the 3D voxel space into empty or occupied voxels, 3D voxel exemplars can capture the 3D shape of objects. Second, viewpoint information is encoded by labeling the occupied voxels into visible or self-occluded voxels. Third, the visible voxels are further classified into truncated or occluded voxels by considering the image borders and other objects in the scene. As a result, 3D voxel exemplars are able to encode information about 3D shape, viewpoint, truncation and occlusion in a uniform 3D space.

3.2. Discovering 3DVPs

A 3DVP represents a group of 3D voxel exemplars which share similar visibility patterns encoded in their 3D voxel models. We discover 3DVPs by clustering 3D voxel exemplars in a uniform 3D space. To do so, we define a similarity score between two 3D voxel exemplars. Formally, a 3D voxel exemplar is represented by a feature vector $\mathbf{x}$ with dimension N^3, where N denotes the size of the 3D voxel space. The elements of the feature vector takes values from a finite set $\mathcal{S} = \{0, 1, 2, 3, 4\}$, which encodes the visibility of the voxels, i.e., 0 for empty voxels, 1 for visible voxels, 2 for self-occluded voxels, 3 for voxels occluded by other objects, and 4 for truncated voxels. Then the similarity metric between two feature vectors $\mathbf{x_1}$ and $\mathbf{x_2}$ is defined as:

$$s(\mathbf{x_1}, \mathbf{x_2}) = \frac{|\mathcal{S}|}{N^3} \sum_{i=1}^{N^3} \mathbb{1}(x_1^i = x_2^i) \cdot w(x_1^i), \quad \text{s.t.,} \sum_{i=0}^{|\mathcal{S}|-1} w(i) = 1, \tag{1}$$

where x_1^i and x_2^i are the ith element of $\mathbf{x_1}$ and $\mathbf{x_2}$ respectively, $\mathbb{1}$ is the indicator function, and $w(i)$ is the weight for voxel status i. The definition in Eq. (1) is general such that the weights can be designed for different applications. For example, if we define all the weights $w(i)$ to $1/5$, the similarity metric in Eq. (1) simply computes the percentage of voxels with the same value. If we use a larger weight for occluded voxels, patterns with similar occluded regions are more likely to be grouped together (See the technical report in [1] for implementation details about the 3D clustering).

After defining the similarity metric between 3D voxel exemplars, we can employ different clustering algorithms in our framework, such as K-means or Affinity Propagation (AP) [14]. Fig. 5 shows several examples of 3D clusters from the KITTI dataset using AP clustering. With the 3D clustering algorithm, we are able to group cars from similar viewpoints and with similar occluded or truncated regions together. We visualize 3DVPs in Fig. 6. For each cluster, we show the 3D voxel model of the cluster center, the average RGB images of the 2D image patches in the cluster, and the average gradient image. Note that there is a high correlation between 3DVP and object appearance including relevant occlusions, which enable us to learn compact and accurate detectors for 3DVPs.

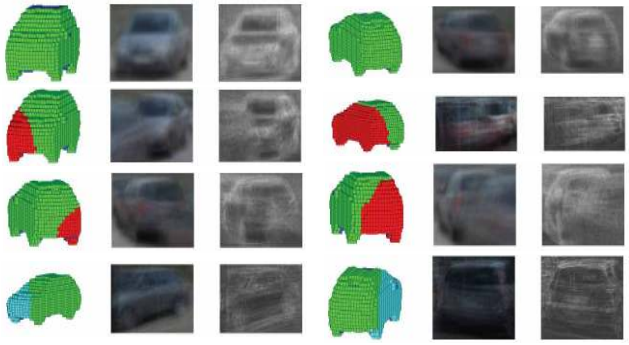


Figure 6. Visualization of selected 3DVPs. We show the 3D voxel model of the cluster center, the average RGB image, and the average gradient image of each 3DVP.

3.3. Learning 3DVP Detectors

We train a detector for each 3DVP with the associated 2D images. Our framework is general to integrate different classifiers in training the detectors, such as support vector machines or boosting. For example, in our experiments, we note that the boosting detector based on Aggregated Channel Features (ACF) [8] is more suitable on the KITTI dataset compared to SVM-based detectors [10, 26].

For an 3DVP that contains occlusion, we incorporate the appearance of the occluder which is inside the 2D bounding box of the occludee in training the 3DVP detector, where the 2D bounding box incorporates occluded area of the occludee. The observation behind it is that occlusions are likely to form certain types of patterns between the occluder and the occludee. For example, in street scenes, cars are likely to occlude each other within specific 3D spatial layout. Such cases include cars parking beside the street, cars lining up on the road, and so on. Incorporating the appearance of the occluder into modeling helps us to detect the occludee by leveraging these occlusion patterns. The 3DVPs we discover in the 3D clustering process capture these occlusion patterns. As we can see from Fig. 5 and Fig. 6, the included regions from the occluders in an occluded 3DVP also share similar appearance, which ensures us to train a reliable detector for the occluded 3DVP. For a truncated 3DVP, image patches corrsponding to the truncated objects are used to train the detector without padding.

3.4. Occlusion Reasoning with 3DVPs

After training all the 3DVP detectors, we can apply them to an input image and obtain the 2D detections. Then, we are able to transfer the meta-data from the 3DVP to the detected objects, which includes the 2D segmentation mask, the 3D pose and the 3D shape as shown in Fig. 2(b). These meta-data enable us to perform a global occlusion reasoning among all the detected objects in the scene, which outputs mutually consistent detections.

Let $\mathbb{D} = \{d_1, d_2, ...\}$ denote the detection hypotheses in an image I, where d_i is a binary variable indicating if the detection hypothesis is true or false. We represent a detection d_i by its detection score s_i, and its 2D visibility mask m_i that are derived from the 3DVP. Specifically, we transfer the 2D segmentation mask associated with the cluster center of the 3DVP to the detection, and rescale it to fit the bounding box of the detection. m_i is composed of three components: m_i^v (visible region), m_i^o (occluded region), and m_i^t (truncated region) (refer to examples in Fig. 4). We design our occlusion reasoning model using an energy-based conditional random field model [23], which favors to have detections that are mutually consistent to each other. Underlying intuition is that 1) all the invisible regions of selected detections shall be explained either by another occluding object or by image truncation, and 2) visible regions of selected detections should not overlap with each other. The model is formulated as:

$$E(\hat{\mathbb{D}}) = \sum_{i\in\hat{\mathbb{D}}} \Big(\underbrace{w_d(s_i - b)}_{\text{detection score}} - \underbrace{w_o \frac{|m_i^o| + |m_i^t|}{|m_i|}}_{\text{invisibility penalty}} + \underbrace{w_o \frac{|m_i^t \not\subset I|}{|m_i|}}_{\text{truncation explained}} \Big) + \sum_{i,j\in\hat{\mathbb{D}}, i\neq j} \Big(\underbrace{w_o \frac{|m^o_{\text{far}(i,j)} \cap m^v_{\text{near}(i,j)}|}{|m_{\text{far}(i,j)}|}}_{\text{occlusion explained}} - \underbrace{w_p \frac{\sum_{k=v,o,t} |m_i^k \cap m_j^k|}{\min(|m_i|, |m_j|)}}_{\text{overlap penalty}} \Big) \tag{2}$$

where w_d, w_o, w_p and b are the model parameters, $|\cdot|$ operator measures the area of a region, far$(\cdot)$ and near$(\cdot)$ return far and near object based on the bottom position of a detection, and $\hat{\mathbb{D}} \subseteq \mathbb{D}$. Our model has a number of favorable properties. First, detection outputs that are associated with largely occluded patterns are penalized by the invisibility penalty term in Eq. (2) unless the occluded area is explained by other objects (see the "occlusion explained" term). Similarly, truncated detections are also penalized by the "invisibility penalty" term unless they are located in accordance with the image boundary (see the "truncation explained" term). Second, detections that overlap largely with other detections are penalized according to the overlap penalty term in Eq. (2), which implements a similar concept as non-maximum suppression, but our model is more fine-grained as it measures the overlap between visible areas.

Solving the exact inference problem of our occlusion reasoning model is infeasible as the graph is often very complex, i.e., there are many overlapping detections which create a locally fully connected graph. So we solve the MAP inference problem with a greedy algorithm. Starting from an empty set $\hat{\mathbb{D}}_0 = \emptyset$, we add one detection d_i to the set $\hat{\mathbb{D}}_k$ in each iteration k that maximizes the energy improvement $E(\hat{\mathbb{D}}_k \cup d_i) - E(\hat{\mathbb{D}}_k)$ until the energy improvement is smaller than zero. In order to rank detections, we compute the posterior marginals from the estimated MAP as in [5], and use them as detection scores. We train the model parameters by grid search on the validation set.

4. Experiments

4.1. Datasets and Evaluation Metrics

Datasets. We apply our object recognition framework to the KITTI detection benchmark [17] and the outdoor-scene (OutdoorScene) dataset [41] for car detection. The KITTI dataset contains 7,481 images for training, and 7,518 images for testing. These are video frames from autonomous driving scenes. We focus on the car category in KITTI, since there are 28,612 cars in the training set which provides enough data to test our data-driven approach. Since the ground truth annotations of the KITTI test set are not released, we split the KITTI training images into train set and validation set to conduct analyses about our framework, which contain 3,682 images and 3,799 images respectively. Our splitting ensures that there is no images from the same video across the train and validation sets. We also evaluate our algorithm on the entire test set. The OutdoorScene dataset contains 200 images from various sources, which is designed to test object detectors in the presence of severe occlusions and truncation. There are 659 cars in total, among which 235 cars are occluded and 135 cars are truncated. This dataset is used for testing only.

Evaluation Metrics. We evaluate our recognition results at the three difficulty levels, easy, moderate, and hard, suggested by the KITTI benchmark [16]. To evaluate the object detection accuracy, the Average Precision (AP) [9] is reported throughout the experiments. 70% overlap threshold is adopted in the KITTI benchmark for car. To evaluate jointly object detection and orientation estimation, [17] proposes a new metric called Average Orientation Similarity (AOS), which is defined as $AOS = \frac{1}{11}\sum_{r\in\{0,0.1,\ldots,1\}} \max_{\tilde{r}:\tilde{r}\geq r} s(\tilde{r})$, where r is the detection recall, and $s(r) \in [0,1]$ is the orientation similarity at r (see [17] for details). In addition, we propose two new evaluation metrics to measure the accuracy of 2D segmentation and 3D localization jointly with detection. For 2D segmentation, we define Average Segmentation Accuracy (ASA) by replacing the orientation similarity in AOS with the 2D pixel segmentation accuracy. For the 3D localization, we define Average Localization Precision (ALP) by replacing orientation similarity in AOS with localization precision, i.e., a 3D location is considered to be correct if its distance from the ground truth 3D location is smaller than certain threshold. For object detection evaluation on the OutdoorScene dataset, we use the standard 50% overlap criteria of PASCAL VOC [9].

4.2. Analysis on KITTI Validation Set

In this section, we present the detailed analysis on our method using our validation split of the KITTI training set.

2D v.s. 3D Clustering. We show that the our method, which discovers 3DVPs with 3D clustering and trains detectors on 3DVPs, can improve the object detection performance compared with its 2D counterpart proposed in the literature [6, 27] that discovers subcategories with 2D clustering and trains detectors on the subcategories. For 2D clustering, we extract visual features from the training instances, and cluster them by computing similarity between the 2D features similarly to [6, 27]. We experiment with two clustering algorithms, K-means and Affinity Propagation (AP) [14], with different numbers of clusters. The control parameter in AP is varied to obtain different number of clusters. We train ACF detectors [8] for both 2D and 3D clusters. Table 1 shows the average precisions by applying the trained ACF detectors to the validation set. We can see from the table that 3D K-means and 3D AP outperform their 2D counterparts significantly. Our evaluation verifies that 3DVP-driven detectors can better capture the appearance variation of an object category compared to the 2D appearance-driven detectors. We also observe that 3D AP is less susceptible to the choice of the cluster numbers. In the following analyses, we experiment with the 3DVP detectors trained on 125 clusters from 3D AP clustering.

	Object Detection (AP)			Orientation (AOS)		
Methods	Easy	Moderate	Hard	Easy	Moderate	Hard
DPM [10] NMS.5	54.91	42.49	32.73	33.71	26.30	20.37
DPM [10] INMS.6	44.35	36.49	28.87	27.45	22.71	18.07
Ours NMS.5	79.06	64.72	50.38	77.65	62.75	48.57
Ours INMS.6	78.28	65.62	54.90	76.87	63.49	52.57
Ours Occlusion	**80.48**	**68.05**	**57.20**	**78.99**	**65.73**	**54.67**

Table 2. AP/AOS comparison between different detection/decoding methods on the validation set. We show the results of 3D AP with 125 clusters for **Ours**.

Decoding Detection Hypotheses. Table 2 compares the detection and the orientation estimation accuracies on the validation set among DPM baselines and our 3DVP detectors using different decoding schemes. As the first decoding scheme, we adopt the popular Non-Maximum Suppression (NMS) implemented by [10]. The method computes the overlap between two bounding boxes by $\frac{|o_i \cap o_j|}{|o_i|}$ and greedily suppresses detections that have larger than 0.5 overlap with already selected ones. Since this method (NMS.5) tends to suppress less confident occluded detections aggressively, which hurts the performance of 3DVP detectors in Hard case, we adopt another NMS method based on Intersection over Union (IoU) $\frac{|o_i \cap o_j|}{|o_i \cup o_j|}$ with 0.6 threshold (INMS.6). It performs the same suppression procedure as NMS.5, but using the 0.6 IoU threshold. INMS.6 tends to keep more occluded detection hypotheses and achieves better performance in moderate and hard cases compared to NMS.5. Finally, our occlusion reasoning method improves the detection and orientation estimation accuracies with significant margins in all difficulty levels. The superior results

2D K-means				3D K-means				2D Affinity Propagation				3D Affinity Propagation			
K	Easy	Moderate	Hard	K	Easy	Moderate	Hard	K	Easy	Moderate	Hard	K	Easy	Moderate	Hard
5	44.21	31.23	25.42	5	41.78	31.63	28.06	137	**46.76**	**35.66**	**32.30**	87	74.28	62.54	52.87
10	47.78	38.13	32.26	10	52.55	39.44	32.76	156	46.12	34.44	30.35	125	78.28	**65.62**	**54.90**
20	61.24	48.04	40.27	20	61.52	49.33	42.07	189	44.97	34.88	31.53	135	78.13	65.44	54.79
30	**67.83**	51.68	43.63	30	63.29	49.46	41.55	227	39.66	31.67	29.62	152	77.96	64.45	53.93
40	66.49	**53.18**	**45.96**	40	69.46	56.13	47.26	273	36.52	28.51	27.08	180	79.02	65.55	54.72
50	66.65	51.90	43.28	50	70.76	58.77	50.30	335	27.96	22.74	22.22	229	79.94	64.87	53.53
100	58.45	46.15	39.34	100	75.73	61.06	51.29					284	79.91	64.04	53.10
150	56.74	43.84	37.75	150	77.15	63.25	53.13					333	**79.98**	63.95	52.99
200	53.57	41.26	33.61	200	78.00	**64.81**	**54.30**								
250	53.86	39.81	33.58	250	76.85	63.48	53.93								
300	48.81	35.53	29.10	300	**78.10**	62.11	51.99								
350	42.68	33.55	27.35	350	74.78	62.00	51.81								

Table 1. AP Comparison between 2D and 3D clustering with k-means and affinity propagation on our validation split. The table shows the average precision obtained by training ACF detectors in different settings.

Method	Easy	Moderate	Hard
Joint 2D Detection and Segmentation (ASA)			
DPM [10]+box	38.09	29.42	22.65
Ours INMS.6+box	57.52	47.84	40.01
Ours Occlusion+box	59.21	49.74	41.71
Ours INMS.6+3DVP	63.88	52.57	43.82
Ours Occlusion+3DVP	**65.73**	**54.60**	**45.62**
Joint 2D Detection and 3D Localization (ALP)			
DPM [10] $<$ 2m	40.21	29.02	22.36
Ours INMS.6 $<$ 2m	64.85	49.97	41.14
Ours Occlusion $<$ 2m	**66.56**	**51.52**	**42.39**
DPM [10] $<$ 1m	24.44	18.04	14.13
Ours INMS.6 $<$ 1m	44.47	33.25	26.93
Ours Occlusion $<$ 1m	**45.61**	**34.28**	**27.72**

Table 3. Comparison between different settings of our method and DPM for the 2D segmentation and 3D localization evaluation on our validation split, where 125 clusters from 3D AP clustering are used for **Ours**.

	Object Detection (AP)			Orientation (AOS)		
Methods	Easy	Moderate	Hard	Easy	Moderate	Hard
ACF [8]	55.89	54.74	42.98	N/A	N/A	N/A
DPM [10]	71.19	62.16	48.43	67.27	55.77	43.59
DPM-VOC+VP [29]	74.95	64.71	48.76	72.28	61.84	46.54
OC-DPM [30]	74.94	65.95	53.86	73.50	64.42	52.40
SubCat [27]	81.94	66.32	51.10	80.92	64.94	50.03
AOG [24]	84.36	71.88	59.27	43.81	38.21	31.53
SubCat [28]	84.14	75.46	59.71	83.41	74.42	58.83
Regionlets [36]	84.75	**76.45**	59.70	N/A	N/A	N/A
Ours INMS.6	84.81	73.02	63.22	84.31	71.99	62.11
Ours Occlusion	**87.46**	75.77	**65.38**	**86.92**	**74.59**	**64.11**

Table 4. AP/AOS Comparison between different methods on the KITTI test set. We show the results of 3D AP with 227 clusters for **Ours**. More comparisons are available at [16].

verifies that 3DVP detectors are able to learn accurate visibility patterns of the objects, which provides reliable cues to reason about the occlusion relationship between objects.

Joint 2D Detection and Segmentation Evaluation. We analyze the accuracy of the transferred 2D segmentation mask from 3DVP in terms of 2D segmentation accuracy. Since the KITTI dataset does not provide the ground truth segmentation masks of the objects, we use the 2D segmentation masks obtained by projecting registered 3D CAD models as the ground truth (Fig. 3). Because the registration is guided by the ground truth 3D annotations, the obtained masks are accurate for evaluation. We use the ASA metric described in Sec. 4.1 for the evaluation. Table 3 shows the accuracies of different methods. As DPM [10] does not provide any segmentation information, we treat the whole region inside the bounding box as the segmentation mask (denoted as +box). As the results demonstrate, our 3DVP induced segmentations (+3DVP) improve 6%, 5% and 4% in each difficulty level compared to our own baselines (+box) and 17%, 25%, 23% compared to the DPM baseline.

Joint 2D Detection and 3D Localization Evaluation. In Table 3, we also evaluate the 3D localization accuracy using the average localization precision (ALP). The 3D location of a 2D detection is computed by minimizing the reprojection error between a oriented mean 3D cuboid and the 2D bounding box of the detection, where the mean 3D cuboid is obtained by averaging the 3D dimensions of all the training objects, and the orientation is estimated by the detection. The re-projection error is the sum of squared errors in width and height between the projected 3D cuboid and the 2D bounding box. So accurate 2D bounding box and 3D pose produce precise 3D localization. We evaluate the performance using two 3D distance thresholds: 1 meter and 2 meters. In both experiments, **Ours** Occlusion achieves better 3D localization results than **Ours** INMS.6, and improves over the DPM baseline by more than 20% in 2-meter ALP and more than 10% in 1-meter ALP. We note that [44] also evaluates 3D localization on KITTI images. However, the method is trained with external images and only tested on 260 KITTI images. We could not directly compare our results with [44]. Please see Fig. 7 for qualitative results using our method on the validation set.

4.3. KITTI Test Set Evaluation

To compare with the state-of-the-art methods on the KITTI dataset, we train 3DVP detectors with all the KITTI training data, and then test our method on the test set. We present the detection and the orientation estimation results in Table 4. The 3DVPs are obtained using AP clustering with 227 clusters. Each 3DVP detector is trained with the ACF detector [8]. We evaluate the **Ours** INMS.6 and **Ours** Occlusion. Thanks to our 3DVP model, **Ours** INMS.6 al-

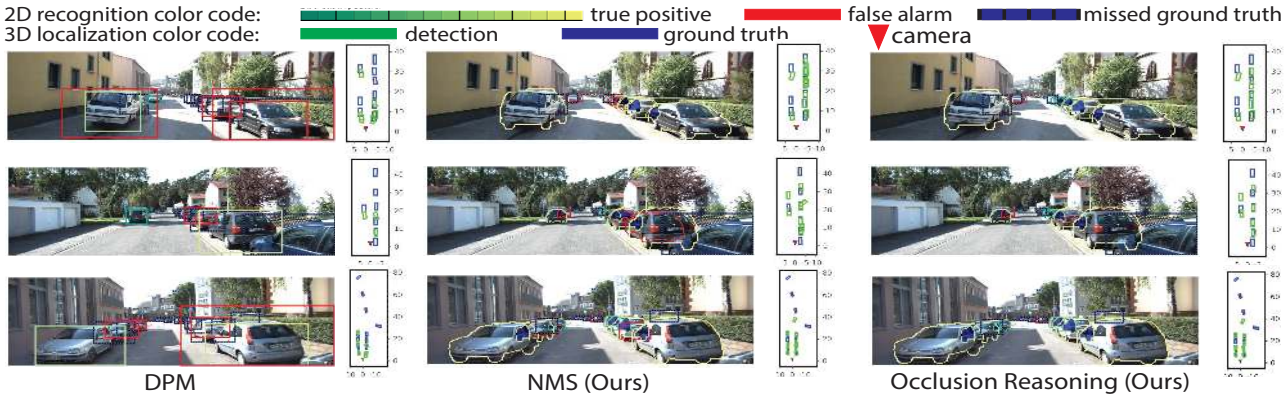


Figure 7. Car recognition results on the KITTI validation set. We compare our method w/wo occlusion reasoning and DPM [10]. Detections at 1 false positive per image (fppi) for the three methods are shown. Blue regions in the images are the estimated occluded areas. Note that severe false alarms in NMS disappear with occlusion reasoning.

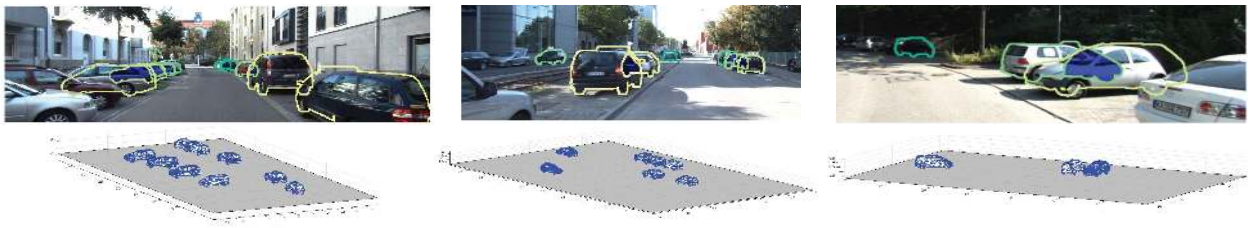


Figure 8. 2D recognition and 3D localization results on the KITTI test set. Blue regions in the images are the estimated occluded areas.

% occlusion	< 0.3	$0.3 - 0.6$	> 0.6
# images	66	68	66
ALM [40]	72.3	42.9	35.5
DPM [10]	75.9	58.6	44.6
SLM [41]	80.2	63.3	52.9
Ours NMS.5	89.7	76.3	55.9
Ours Occlusion	**90.0**	**76.5**	**62.1**

Table 5. AP of the car detection on the OutdoorScene dataset [41].

ready achieves the highest accuracies in most of the difficulty levels for both detection and orientation evaluation. Our full model achieves even further improvement leveraging on the contextual occlusion relationship among objects. The large improvement in the *Hard* category ($\sim 6\%$ compared to the second best method in both object detection and joint detection and orientation estimation) verifies that our algorithm is capable of detecting challenging occluded objects. Notice that SubCat [27] uses the same ensemble of ACF [8] detectors, but using 2D features in clustering. Fig. 8 shows some 2D recognition and 3D localization results on the KITTI test set (see [1] for additional results).

4.4. Object Detection on the OutdoorScene Dataset

We apply our 227 3DVP detectors trained on the whole KITTI training set to the OutdoorScene dataset, and evaluate the object detection accuracy. Since the training and testing images are from different sources, we can test how well our 3DVP detectors trained on the KITTI dataset generalize to other scenarios, such as city and parking lot scenes in the OutdoorScene dataset. Table 5 shows the average precisions for car detection on the dataset, where the test images are partitioned into three different sets according to the amount of occlusion. Our 3DVP detectors outperform ALM [40], DPM [10] and SLM [41] on all the three partitions, which demonstrates the generalization capability of our 3DVP detectors. Similarly to our KITTI experiments, our occlusion reasoning algorithm further improves the detection accuracy in the largely occluded test set.

5. Conclusion

We have proposed a novel 3D object representation, *3D Voxel Pattern*, that enables us to estimate detailed properties of objects beyond 2D bounding boxes, identify challenging occluded objects, and reason about the occlusion relationship between objects. The experimental evaluation demonstrates that our method can recognize objects in complex scenes with high accuracy, while providing detailed 2D/3D properties of the objects. The proposed occlusion reasoning method empowered by the properties further improves the recognition accuracy in various tasks. In addition, the experiment on the OutdoorScene dataset confirms that our model generalizes well to different scenarios. Although the framework is evaluated on the "car" category, we believe that the idea of 3DVP is applicable to generic rigid object categories. We consider generalize the method toward other rigid object categories as a future direction.

Acknowledgments

We acknowledge the support of NSF CAREER grant N.1054127, ONR award N000141110389, and DARPA UPSIDE grant A13-0895-S002.

References

[1] 3dvp. http://cvgl.stanford.edu/projects/3DVP. 4, 8

[2] Trimble 3d warehouse. http://3dwarehouse.sketchup.com. 3

[3] L.-C. Chen, S. Fidler, and R. Urtasun. Beat the mturkers: Automatic image labeling from weak 3d supervision. In *CVPR*, 2014. 4

[4] X. Chen, A. Shrivastava, and A. Gupta. Enriching visual knowledge bases via object discovery and segmentation. In *CVPR*, 2014. 2, 3

[5] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *IJCV*, 95(1):1–12, 2011. 5

[6] S. K. Divvala, A. A. Efros, and M. Hebert. How important are deformable parts in the deformable parts model? In *ECCVW*, pages 31–40, 2012. 2, 3, 6

[7] S. K. Divvala, A. Farhadi, and C. Guestrin. Learning everything about anything: Webly-supervised visual concept learning. In *CVPR*, 2014. 2, 3

[8] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *TPAMI*, 36(8):1532–1545, 2014. 5, 6, 7, 8

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 6

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010. 1, 2, 5, 6, 7, 8

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005. 2

[12] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, pages II–264, 2003. 2

[13] S. Fidler, S. Dickinson, and R. Urtasun. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NIPS*, pages 611–619, 2012. 2

[14] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *science*, 315(5814):972–976, 2007. 4, 6

[15] T. Gao, B. Packer, and D. Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, pages 1361–1368, 2011. 2

[16] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Kitti object detection benchmark. http://www.cvlibs.net/datasets/kitti/eval_object.php. Accessed: 2015-03-18. 6, 7

[17] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, pages 3354–3361, 2012. 1, 2, 3, 4, 6

[18] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *arXiv preprint arXiv:1311.2524*, 2013. 1, 3

[19] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich. Viewpoint-aware object detection and pose estimation. In *ICCV*, pages 1275–1282, 2011. 2

[20] M. Hejrati and D. Ramanan. Analyzing 3d objects in cluttered images. In *NIPS*, pages 593–601, 2012. 2

[21] D. Hoiem, C. Rother, and J. Winn. 3d layoutcrf for multi-view object class recognition and segmentation. In *CVPR*, 2007. 2

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1, 3

[23] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. Huang. A tutorial on energy-based learning. *Predicting structured data*, 2006. 5

[24] B. Li, T. Wu, and S.-C. Zhu. Integrating context and occlusion for car detection by hierarchical and-or model. In *ECCV*, pages 652–667. 2014. 1, 7

[25] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3d feature maps. In *CVPR*, 2008. 2

[26] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, pages 89–96, 2011. 3, 5

[27] E. Ohn-Bar and M. M. Trivedi. Fast and robust object detection using visual subcategories. In *CVPRW*, pages 179–184, 2014. 2, 3, 6, 7, 8

[28] E. Ohn-Bar and M. M. Trivedi. Learning to detect vehicles by clustering appearance patterns. *T-ITS*, 2015. 7

[29] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Multi-view and 3d deformable part models. *TPAMI*, 2015. 7

[30] B. Pepikj, M. Stark, P. Gehler, and B. Schiele. Occlusion patterns for object class detection. In *CVPR*, pages 3286–3293, 2013. 1, 2, 7

[31] S. Savarese and L. Fei-Fei. 3d generic object categorization, localization and pose estimation. In *ICCV*, 2007. 2

[32] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *ICCV*, 2009. 2

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

[34] A. Vedaldi and A. Zisserman. Structured output regression for detection with partial truncation. In *NIPS*, pages 1928–1936, 2009. 2

[35] X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, pages 32–39, 2009. 2

[36] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, pages 17–24, 2013. 7

[37] C. Wojek, S. Walk, S. Roth, and B. Schiele. Monocular 3d scene understanding with explicit occlusion reasoning. In *CVPR*, pages 1993–2000, 2011. 2

[38] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *CVPR*, volume 1, pages 90–97, 2005. 2

[39] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In *WACV*, 2014. 4

[40] Y. Xiang and S. Savarese. Estimating the aspect layout of object categories. In *CVPR*, 2012. 2, 8

[41] Y. Xiang and S. Savarese. Object detection by 3d aspectlets and occlusion reasoning. In *ICCVW*, pages 530–537, 2013. 1, 2, 6, 8

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

[43] M. Z. Zia, M. Stark, and K. Schindler. Explicit occlusion modeling for 3d object class representations. In *CVPR*, pages 3326–3333, 2013. 2

[44] M. Z. Zia, M. Stark, and K. Schindler. Are cars just 3d boxes?–jointly estimating the 3d shape of multiple objects. In *CVPR*, 2014. 7