Learning And-Or Model to Represent Context and Occlusion for Car Detection and Viewpoint Estimation

Tianfu Wu*, Bo Li* and Song-Chun Zhu

Abstract—This paper presents a method for learning an And-Or model to represent context and occlusion for car detection and viewpoint estimation. The learned And-Or model represents car-to-car context and occlusion configurations at three levels: (i) spatially-aligned cars, (ii) single car under different occlusion configurations, and (iii) a small number of parts. The And-Or model embeds a grammar for representing large structural and appearance variations in a reconfigurable hierarchy. The learning process consists of two stages in a weakly supervised way (i.e., only bounding boxes of single cars are annotated). Firstly, the structure of the And-Or model is learned with three components: (a) mining multi-car contextual patterns based on layouts of annotated single car bounding boxes, (b) mining occlusion configurations between single cars, and (c) learning different combinations of part visibility based on CAD simulations. The And-Or model is organized in a directed and acyclic graph which can be inferred by Dynamic Programming. Secondly, the model parameters (for appearance, deformation and bias) are jointly trained using Weak-Label Structural SVM. In experiments, we test our model on four car detection datasets — the KITTI dataset [1], the PASCAL VOC2007 car dataset [2], and two self-collected car datasets, namely the Street-Parking car dataset and the Parking-Lot car dataset, and three datasets for car viewpoint estimation — the PASCAL VOC2006 car dataset [2], the 3D car dataset [3], and the PASCAL3D+ car dataset [4]. Compared with state-of-the-art variants of deformable part-based models and other methods, our model achieves significant improvement consistently on the four detection datasets, and comparable performance on car viewpoint estimation.

Index Terms—Car Detection, Car Viewpoint Estimation, And-Or Graph, Hierarchical Model, Context, Occlusion Modeling.

1 Introduction

1.1 Motivation and Objective

Car is one of the most frequently seen object category in every day scenes. Car detection and viewpoint estimation by a computer vision system has broad applications such as autonomous driving and parking management. Fig. 1 shows a few examples with varying complexities in car detection from four datasets. Car detection and viewpoint estimation are challenging problems due to the large structural and appearance variations, especially ubiquitous occlusions which further increase the intra-class variations significantly. In this paper, we are interested in learning a unified model which can detect cars in the four datasets and estimate car viewpoints. We aim to address two main issues in the following.

The first is to explicitly represent occlusion. Occlusion is a critical aspect in object detection for several reasons: (i) we do not know ahead of time what portion of an object (e.g. car) will be visible in a test image; (ii) we also do not know the occluded areas in weakly-labeled training data (i.e. only bounding boxes of single cars are given, as considered in this paper); and (iii) object occlusions in testing data could be very different from those in training data. Handling occlusions entails models capable of capturing the underlying regularities of occlusions at part level (i.e. different occlusion configurations).

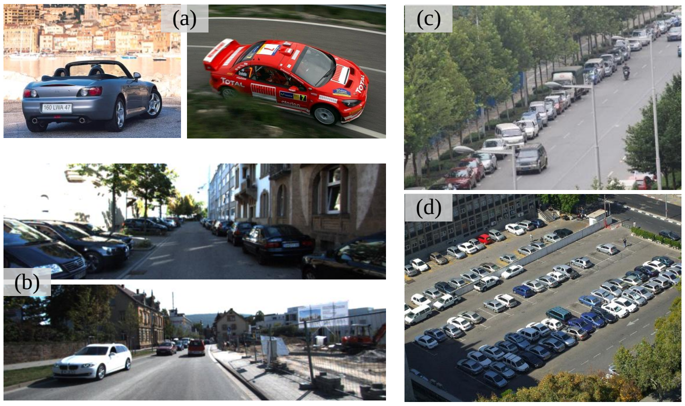


Fig. 1. Illustration of varying complexities in car detection from four datasets. (a) The PASCAL VOC2007 car dataset [2] consists of single cars under different viewpoints but with less occlusion as pointed out in [5]. (b) The KITTI car benchmark [1] includes on-road cars captured by a camera mounted upon a driving car which have more occlusions but restricted viewpoints. (c) The Street-Parking car dataset [6] includes cars with heavy occlusions but less multi-car context and (d) The Parking-Lot car dataset [7] consists of cars with heavy occlusions and rich multi-car context. The proposed And-Or model is learned for car detection in all four datasets.

The second is to explicitly exploit contextual information co-occurring with occlusions (see examples in Fig.1 (b), (c) and (d)), which goes beyond single-car detection. We focus on car-to-car contextual patterns (e.g., different multi-car configurations such as $2, 3$ or 4 cars), which will be utilized in detection and viewpoint estimation and naturally integrated with occlusion configurations.

To represent both occlusion and context, we propose to

- T.F. Wu is with the Department of Statistics, University of California, Los Angeles. E-mail: tfwu@stat.ucla.edu
- B. Li is with Beijing Lab of Intelligent Information Technology, Beijing Institute of Technology, China and a visiting student at University of California, Los Angeles. E-mail: boli86@bit.edu.cn
- S.-C. Zhu is with the Department of Statistics and Computer Science, University of California, Los Angeles. E-mail: sczhu@stat.ucla.edu
- * Equal contribution.

Manuscript received MM DD, YYYY; revised MM DD, YYYY.

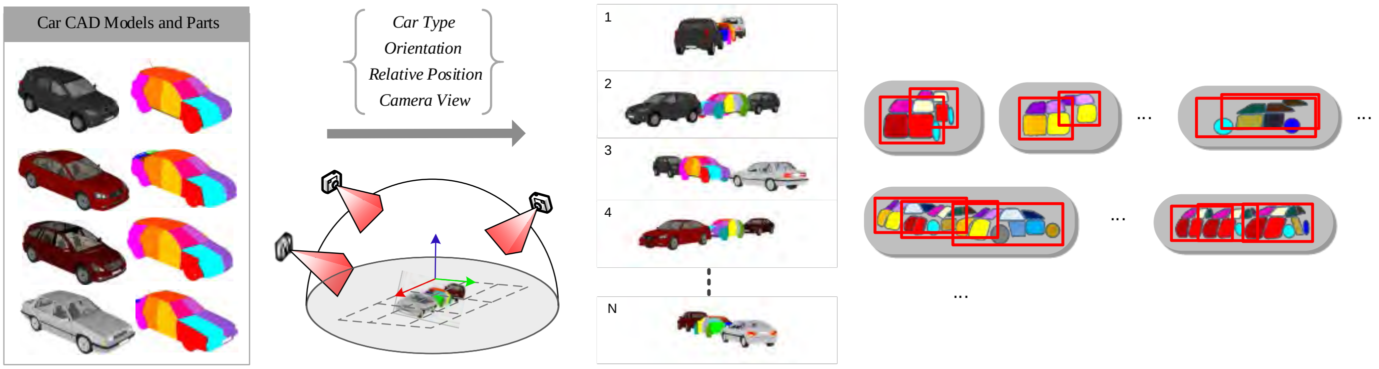


Fig. 2. Illustration of the statistical regularities of car occlusions and multi-car contextual patterns by CAD simulation. We represent car-to-car occlusion at semantic part level (left) and generate a large number of synthetic occlusion configurations (middle) w.r.t. four factors (car type, orientation, relative position and camera view). We represent the regularities of different combinations of part visibilities (i.e., occlusion configurations) by a hierarchical And-Or model. This model also represents multi-car contextual patterns (right) based on the geometric configurations of single cars.

learn an And-Or model which takes into account structural and appearance variations at multi-car, single-car and part levels jointly. Our And-Or model belongs to grammar models [8], [9] embedded in a hierarchical graph structure, which can express a large number of configurations (occlusion configurations and multi-car configurations) in a compositional and reconfigurable manner. Fig.3 illustrates our And-Or model. By reconfigurable, it means that we learn appearance templates and deformation models for single cars and parts, and the composed appearance templates for a multi-car contextual pattern is inferred on-the-fly in detection according to the selections of their child single car Or-nodes. So, our model can express a large number of multi-car contextual patterns with different compatible occlusion configurations of single cars. *Reconfigurability* is one of the most desired property in hierarchical models, which plays the main role in boosting the performance in our experiments, and also distinguishes the proposed method to other models such as the visual phrase model [10] and different object-pair models [11], [12], [13], [14].

1.2 Method Overview

1.2.1 Data Preparation with Simulation Study

Manually annotating car views, parts and part occlusions on real images are time-consuming and usually error-prone. One innovation in this paper is that we generate a large set of occlusion configurations and multi-car configurations by CAD models [1] and a publicly available graphics rendering engine, the SketchUp SDK [2]. In the CAD simulation, the occlusion configurations and multi-car contextual patterns reflect variations in four factors: *car type, orientation, relative position and camera view*. We decompose a car into 17 semantic parts as shown in different colors in the left side of Fig. 2. We then generate a large number of examples by placing 3 cars in a 3×3 grid (resembling the regularities of cars in parking lots or on the road, see the middle of Fig. 2). For the cars in the center, we compare their part visibilities from different viewpoints (as illustrated by the camera icons), and obtain *the part occlusion data matrix* (each row represents an example and each entry takes a binary value, 0/1, representing occluded or not for a part under a viewpoint). The data matrix is used to learn the occlusion configurations. Similarly, we learn different multi-car contextual patterns based on the geometric configurations (see some examples in the right side of Fig. 2). Note that the semantic part annotations in the synthetic examples are used to learn the structure of our And-Or model and the parts are treated as latent variables in weakly-annotated training data of real images. We do not evaluate the performance of part localization and instead evaluate the viewpoint estimation based on the inferred part configurations.

In the simulation, we place 3 cars in a 3×3 grid with three considerations: (i) It can generate different occlusion configurations for the car in the center under different camera viewpoints, as well as different multi-car contextual patterns (2-car or 3-car pattern), which is easier than using 2 cars in processing the data in simulation. (ii) It can generate the synthetic dataset in which the occlusion configurations and multi-car contextual patterns are generic enough to cover the four situations in Fig.1. (iii) It can also reduce the gap between the synthetic data and real data when learning the initial appearance parameters for parts with the car in the back instead of the white background (see more details in Sec.5).

1.2.2 The And-Or Model

There are three types of nodes in the And-Or model: an *And-node* represents decomposition (e.g., a car is composed of a small number of parts), an *Or-node* represents alternative ways of decomposition accounting for structural variations (e.g., different part configurations of a single car due to occlusions), and a *Terminal-node* captures appearance variations to ground a car or a part to image data.

Fig. 3 illustrates the learned And-Or model. The hierarchy consists of a layer of multi-car contextual patterns (top) and several layers of occlusion configurations of single cars (bottom). The overall structure is as-follows:

i) The root Or-node represents different multi-car configurations which capture both viewpoints and car-to-car contextual patterns. Each multi-car contextual pattern is then represented by an And-node (e.g., car pairs and car triples shown in the figure). The contextual information reflect the layout regularities of a small number, N (e.g.,

[1] we used 40 CAD models selected from www.doschdesign.com and Google 3D warehouse

[2] www.sketchup.com

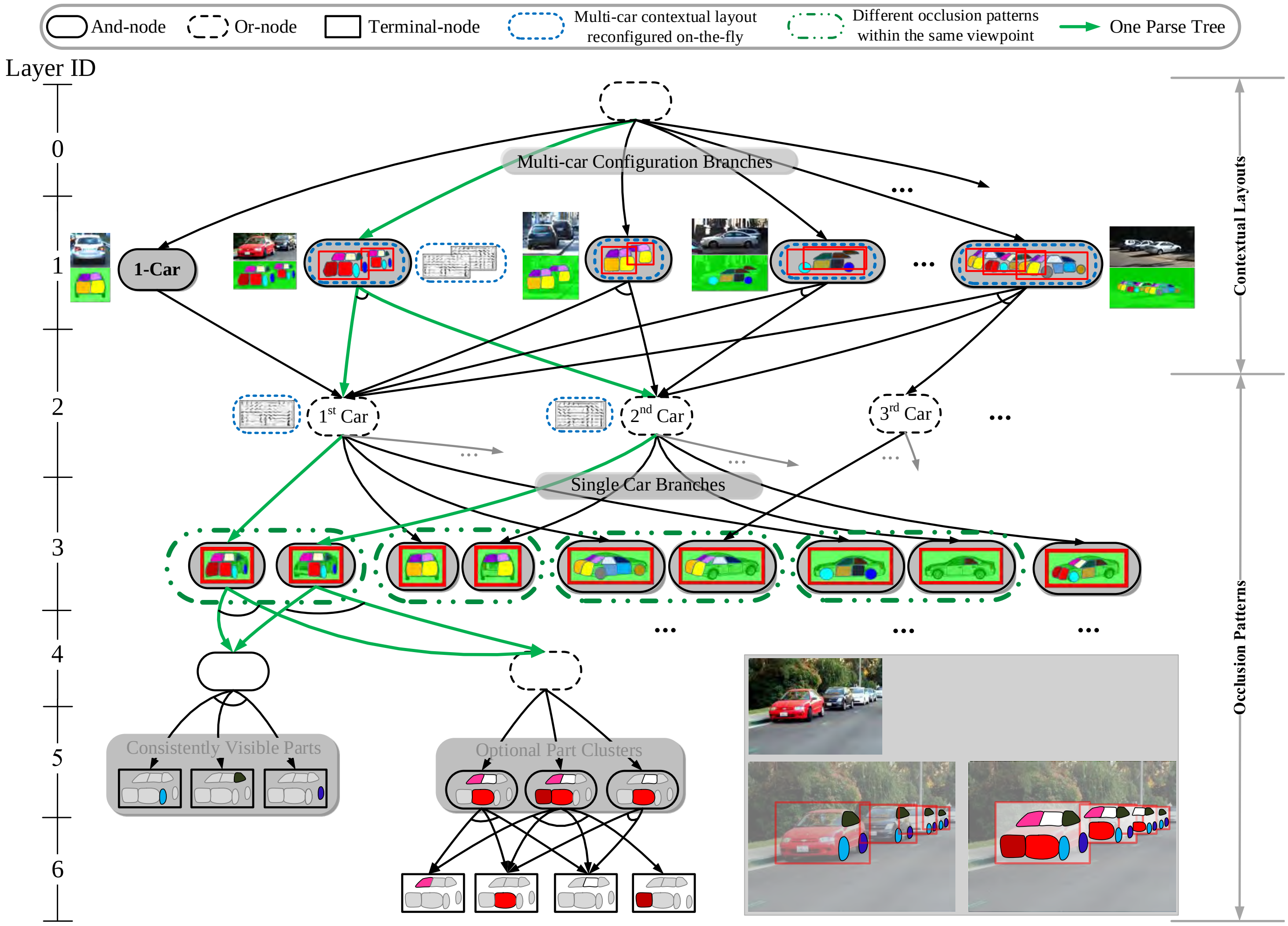

And-node
Or-node
Terminal-node
Multi-car contextual layout reconfigured on-the-fly
Different occlusion patterns within the same viewpoint
One Parse Tree
Layer ID
0
1
2
3
4
5
6
Multi-car Configuration Branches
1-Car
1st Car
2nd Car
3rd Car
Single Car Branches
Consistently Visible Parts
Optional Part Clusters
Contextual Layouts
Occlusion Patterns

Fig. 3. Illustration of our And-Or model for car detection. It represents multi-car contextual patterns and occlusion configurations jointly by modeling spatially-aligned multi-cars together and composing visible parts explicitly for single cars. (Best viewed in color)

$N \in \{2,3\}$), of cars in real sitations (such as cars in a parking lot).

ii) A multi-car And-node is decomposed into nodes representing single cars. Each single car is represented by an Or-node (e.g., the 1^{st} car and the 2^{nd} car), since we have different combinations of car types, viewpoints and occlusion configurations.Here, a multi-car And-node embeds the reconfigurable compositional grammar of a multi-car configuration (e.g., the three 2-car configurations in the right-top of Fig.2) in which the single cars are reconfigurable w.r.t. viewpoint and occlusion configuration (up to some extend), and car type. This reconfigurability gives our model expressive power to handle the large variations of multi-car configurations in real sitations.

iii) Each occlusion configuration is represented by an And-node which is further decomposed into parts. Parts are learned using CAD simulation (i.e., the 17 semantic parts) and are organized into consistently visible parts and optional part clusters (see the example in the right-bottom of Fig. 3). Then, a single car can be represented by the consistently visible parts (i.e., And) and one of the optional part clusters (i.e., Or). The green dashed bounding boxes show some examples corresponding to different occlusion configurations (i.e., visible parts) from the same viewpoint.

1.2.3 Weakly-supervised Learning of the And-Or Model

Using weakly-annotated real image training data and the synthetic data, we learn the And-Or model in two stages:

i) Learning the structure of the hierarchical And-Or model. Both the multi-car contextual patterns and occlusion configurations of single cars are learned automatically based on the annotated single car bounding boxes in training data together with the synthetic examples generated from CAD simulations. The multi-car contextual patterns are mined or clustered from the geometric layout features. The occlusion configurations are learned by a clustering method using the part visibility data matrix. The learned structure is a directed and acyclic graph since we have both single-car-sharing and part-sharing, thus Dynamic Programming (DP) can be applied in inference.

ii) Learning the parameters for appearance, deformation and bias. Given the learned structure of the And-Or model, we jointly train the parameters in the structural SVM framework and adopt the Weak-Label Structural SVM (WLSSVM) method [15], [16] in implementation.

1.2.4 Experiments

In experiments, we evaluate the detection performance of our model on four car datasets: the KITTI dataset [1], the

PASCAL VOC2007 car dataset [2] and two self-collected datasets – the Street-Parking dataset [6] and the Parking Lot dataset [7] (which are released with this paper). Our model outperforms different state-of-the-art variants of DPM [17] (including the latest implementation [18]) on all the four datasets, as well as other state-of-the-art models [6], [14], [19], [20] on the KITTI and the Street-Parking datasets. We evaluate viewpoint estimation performance on three car datasets: the PASCAL VOC2006 car dataset [2], the 3D car dataset [3], and the PASCAL3D+ car dataset [4]. Our model achieves comparable performance with the state-of-the-art methods (significantly better than the method using deep learning features [21]). *The detection code and data are available on the author's homepage* [3].

Paper Organization. The remaining of this paper is organized as follows. Section 2 overviews the related work and summarizes our contributions. Section 3 presents the And-Or model and defines its scoring functions. Section 4 presents the method of mining multi-car contextual patterns and occlusion configurations of single cars in weakly-labeled training data. Section 5 discusses the learning of model parameters using WLSSVM, as well as details of the DP inference algorithm. Section 6 presents the experimental results and comparisons of the proposed model on the four car detection datasets and the three viewpoint estimation datasets. Section 7 concludes the paper with discussions.

2 Related Work and Our Contributions

Over the last decade, object detection has made much progress in various vision tasks such as face detection [22], pedestrian detection [23], and generic object detection [2], [17], [24]. In this section we focus on occlusion and context modeling in object detection, and classify the recent literature into three research streams. For a full review of contemporary approaches, we refer the reader to recent survey articles [25], [26], [27].

i) Single Object Modeling and Occlusion Modeling. Hierarchical models are widely used in the recent literature of object detection and most existing approaches are devoted to learning a single object model. Many work extended the deformable part-based model [17] (which has a two-layer structure) by exploring deeper hierarchy and global part configurations [15], [24], [28], using strong manually-annotated parts [29] or CAD models [30], or keeping human in-the-loop [31]. To address the occlusion problem, various occlusion models estimate the visibilities of parts from image appearance, using assumptions that the visibility of a part is (a) independent from other parts [32], [33], [34], [35], [36], (b) consistent with neighboring parts [15], [37], or (c) consistent with its parent or child parts describing object appearance at different scales [38]. Another essential problem is to organize part configurations. Recently, [6], [15], [34] explored different ways to deal with this problem. In particular, [34] modeled different part configurations by the local part mixtures. [15] used a more flexible grammar model to infer both the occluder and visible parts of an occluded person. [6] regularized parts into consistently visible parts and optional part clusters, which is more efficient to represent occlusion configurations. Recent work [39], [40], [41], [42], [43] proposed to enumerate possible occlusion configurations and model each occlusion configuration as a specific component. [44] proposed a 2D model to learn discriminative subcategories, and [45] further integrated it with an explicit 3D occlusion model, both showing excellent performance on the KITTI dataset. Though those models were successful in some heavily occluded cases, they did not represent contextual information, and usually learned another separate context model using the detection scores as input features. Recently, an And-Or quantization method was proposed to learn And-Or tree models [24], [46] for generic object detection in PASCAL VOC [2] and learn 3D And-Or models [47] respectively, which could be useful in occlusion modeling.

ii) Object-Pair and Visual Phrase Models. To account for the strong co-occurrence, object-pair [11], [12], [13], [14] and visual phrase [10] methods modeled occlusions and interactions using a X-to-X or X-to-Y composite template that spans both one object (i.e., "X" such as a person or a car) and another interacting object (i.e., "X" or "Y" such as the other car in a car-pair in parking lots or a bicycle on which a person is riding). Although these models can handle occlusion better than single object models, the object-pair or visual phrase modeled occlusion implicitly, and they were often manually designed with fixed structures (i.e., not reconfigurable in inference). They performed worse than original DPM in the KITTI dataset as evaluated by [14].

iii) Context Models. Many context models have been exploited in object detection with improved performance [48], [49], [50], [51], [52]. Hoiem et al. [50] explored a scene context, Desai et al. [49] improved object detectors by incorporating the multi-class context on the pascal dataset [2] in a max-margin framework. In [51], Tu and Bai integrated the detector responses with background pixels to determine the foreground pixels. In [52], Chen et. al. proposed a multi-order context representation to take advantage of the co-occurrence of different objects. Recently, [53] explored geographic contextual information to facilitate car detection, and [54] explored a 3D panoramic context in object detection. Although these work verified that context is crucial in object detection, most of them modeled objects and context separately, not in a unified framework.

This paper is extended from our two previous conference papers [6], [7] in the following aspects: (i) A unified representation is learned for integrating occlusion and context; (ii) More details on the learning algorithm and the detection algorithm are presented; (iii) More analyses and comparisons on the experimental results are added with improved performance.

This paper makes three contributions to the literature of car detection.

i) It proposes an And-Or model to represent multi-car context and occlusion configurations. The proposed model is multi-scale and reconfigurable to account for large structure, viewpoint and occlusion variations.

ii) It presents a simple, yet effective, approach to mine context and occlusion configurations from weakly-labeled training data.

iii) It introduces two datasets for evaluating occlusion and multi-car context, and obtains performance comparable

[3] http://www.stat.ucla.edu/~tfwu/projects.htm

to or better than state-of-the-art car detection methods in four challenging datasets.

3 Representation and Inference

3.1 The And-Or Model and Scoring Functions

In this section, we introduce the notations in defining the And-Or model and its scoring functions.

An *And-Or model* is defined by a 3-tuple, $\mathcal{G} = (\mathcal{V}, E, \Theta)$, where $\mathcal{V} = \mathcal{V}_{\text{And}} \cup \mathcal{V}_{\text{Or}} \cup \mathcal{V}_T$, represents the nodes in three subsets: And-nodes $\mathcal{V}_{\text{And}}$, Or-nodes $\mathcal{V}_{\text{Or}}$ and Terminal-nodes $\mathcal{V}_T$; E is the set of edges organizing all the nodes in a directed and acyclic graph (DAG); $\Theta = (\Theta^{app}, \Theta^{def}, \Theta^{bias})$, is the set of parameters (for appearance, deformation and bias respectively, to be defined later).

A *Parse Tree* is an instantiation of the And-Or model by selecting the best child (according to the scoring functions to be defined) for each encountered Or-node. The green arrows in Fig. 3 show an example of parse tree.

Appearance Features. We adopt the Histogram of Oriented Gradients (HOG) feature [17], [55] to describe appearance. Let I be an image defined on an image lattice. Denote by $\mathcal{H}$ the HOG feature pyramid computed for I using λ levels per octave, and by Λ the lattice of the whole pyramid. Let $p = (l, x, y) \in \Lambda$ specify a position (x, y) in the l-th level of the pyramid $\mathcal{H}$. Denote by $\Phi^{app}(\mathcal{H}, p_t)$ the extracted HOG features for a Terminal-node t placing at position p_t in the pyramid.

Deformation Features. We allow local deformation when composing the child nodes into a parent node. In our model, parts are placed at twice the spatial resolution w.r.t. single cars, while single cars and composite multi-cars are at the same spatial resolution. We penalize the displacements between the anchor locations of child nodes (w.r.t. the placed parent node) and their actual deformed locations. Denote by $\delta = [dx, dy]$ the displacement. The deformation feature is defined by,

$$\Phi^{def}(\delta) = [dx^2, dx, dy^2, dy]'.$$

A **Terminal-node** $t \in \mathcal{V}_T$ grounds a single car or a part to image data (see Layer 3 and 4 in Fig.3). Given a parent node A, the model for t is defined by a 4-tuple

$$(\theta_t^{app}, s_t, a_{t|A}, \theta_{t|A}^{def})$$

where $\theta_t^{app} \subset \Theta^{app}$ is the appearance template, $s_t \in \{0, 1\}$ the scale factor for placing node t w.r.t. its parent node, $a_{t|A}$ a two-dimensional vector specifying an anchor position relative to the position of parent node A, and $\theta_{t|A}^{def} \subset \Theta^{def}$ the deformation parameters. Given the position $p_A = (l_A, x_A, y_A)$ of the parent node A, the scoring function of a Terminal-node t is defined by,

$$score(t|A, p_A) = \max_{\delta \in \Delta}(< \theta_t^{app}, \Phi^{app}(\mathcal{H}, p_t) > - \\ < \theta_{t|A}^{def}, \Phi^{def}(\delta) >), \qquad (1)$$

where Δ is the space of deformation (i.e., the lattice of the corresponding level in the feature pyramid), $p_t = (l_t, x_t, y_t)$ with $l_t = l_A - s_t\lambda$ and $(x_t, y_t) = 2^{s_t}(x_A, y_A) + a_{t|A} + \delta$ where $s_t = 0$ means the object and parts are placed at the same resolution and $s_t = 1$ means parts are placed at twice the resolution of the object templates, and $< \cdot, \cdot >$ denotes the inner product. Fig.3 shows some learned appearance templates.

An **And-node** $A \in \mathcal{V}_{\text{And}}$ represents a decomposition of a large entity (e.g., a multi-car layout at Layer 1 or a single car at Layer 3 in Fig.3) into its constituents (e.g., 2 or 3 single cars or a small number of parts). Single car And-nodes are associated with viewpoints. Unlike the Terminal-nodes, single car And-nodes are not allowed to be deformable in a multi-car configuration in this paper (we implemented it in experiments and did not observe performance improvement, so for simplicity we make them not deformable). Denote by $ch(v)$ the set of child nodes of a node $v \in \mathcal{V}_{\text{And}} \cup \mathcal{V}_{\text{Or}}$. The position p_A of an And-node A is inherited from its parent Or-node, and then the scoring function is defined by,

$$score(A, p_A) = \sum_{v \in ch(A)} score(v|A, p_A) + b_A \qquad (2)$$

where $b_A \in \Theta^{bias}$ is the bias term. Each single car And-node (at Layer 3) can be treated as the DPM [17] or the And-Or structure proposed in [6]. So, our model is flexible to integrate state-of-the-art single object models. For multi-car And-nodes (at Layer 1), their child nodes are Or-nodes and the scoring function $score(v|A, p_A)$ is defined below.

An **Or-node** $O \in \mathcal{V}_{\text{Or}}$ represents different structure variations (e.g., the root node and the i-th car node at Layer 2 in Fig.3). For the root Or-node O, when placing at the position $p \in \Lambda$, the scoring function is defined by,

$$score(O, p) = \max_{v \in ch(O)} score(v, p), \qquad (3)$$

where $ch(O) \subset \mathcal{V}_{\text{And}}$. For the i-th car Or-node O, given a parent multi-car And-node A placed at p_A, the scoring function is then defined by,

$$score(O|A, p_A) = \max_{v \in ch(O)} \max_{\delta \in \Delta}(score(v, p_v) - \\ < \theta_{O|A}^{def}, \Phi^{def}(\delta) >), \qquad (4)$$

where $p_v = (l_v, x_v, y_v)$ with $l_v = l_A$ and $(x_v, y_v) = (x_A, y_A) + \delta$. The best child of an Or-node is computed by taking argmax of Eqn.(3) and Eqn.(4).

3.2 The DP Algorithm in Detection

In detection, we place the And-Or model at all positions $p \in \Lambda$ and retrieve the optimal parse trees for all positions at which the scores are greater than the detection threshold. Thanks to the directed and acyclic structure of our And-Or model, we can utilize the efficient DP algorithm which consists of two stages:

In the bottom-up pass: Following the depth-first-search (DFS) order of nodes in the And-Or model, the bottom-up pass computes the matching scores of all possible parse trees of the And-Or model at all possible positions in the whole feature pyramid.

First of all, we compute the appearance score maps (pyramid) for all Terminal-nodes (which is done by filter convolution). The optimal position of a Terminal-node w.r.t. a parent node can be computed as a function of the position of the parent node. The quality (matching score) of the optimal position for a Terminal-node w.r.t. a given position of the parent is computed using Eqn.1 (which yields the deformed score map through the generalized distance transform trick as done in the DPM [17] for efficiency), and the optimal position can be retrieved by replacing $\max$ in Eqn.(1) with $\arg\max$.

Then, following the DFS order of nodes, we compute the score maps for all the And-nodes and Or-nodes using Eqn.(2), (3) and (4) with the score maps of their child nodes having been computed already. Similarly, we can obtain the optimal branch for each Or-node by replacing the $\max$ in Eqn.(3) and (4) with $\arg\max$.

In the top-down pass, we first find all detection candidates for the root Or-node O based on its score maps, i.e., the positions $\mathbb{P} = \{p; score(O,p) \geq \tau \text{ and } p \in \Lambda\}$. Then, following the breadth-first-search (BFS) order of nodes, we retrieve the optimal parse tree at each $p \in \mathbb{P}$: starting from the root Or-node, we select the optimal branch of each encountered Or-node, keep all the child nodes of each encountered And-node, and retrieve the optimal position of each Terminal-node. Based on the parsed sub-tree rooted at single car And-nodes, we obtain the viewpoint estimation and the occlusion configuration.

Post-processing. To generate the final detection results of single cars for evaluation, we apply multi-car guided non-maximum suppression (NMS) to deal with occlusions:

i) Some of the single cars in a multi-car detection candidate are highly overlapped due to occlusion, so if we directly use conventional NMS, we will miss the detection of the occluded cars. We enforce that all the single car bounding boxes in a multi-car prediction will not be suppressed by each other. A similar idea is also used in [12].

ii) Overlapped multi-car detection candidates might report multiple predictions for the same single car. For example, if a car is shared by a 2-car detection candidate and a 3-car detection candidate, it will be reported twice. We will keep only the one with higher score.

4 Learning And-Or Structures

In this section, we present the methods of learning the structures of And-Or model by mining contextual patterns and occlusion configurations in the positive training dataset.

4.1 Generating Multi-car Training Samples

Positive Samples. Denote by $D^+ = \{(I_1, \mathbb{B}_1), \cdots, (I_n, \mathbb{B}_n)\}$ the positive training dataset with $\mathbb{B}_i = \{B_i^j = (x_i^j, y_i^j, w_i^j, h_i^j)\}_{j=1}^{k_i}$ being the set of k_i annotated single car bound boxes in image I_i. Here, (x, y) is the left-top corner and (w, h) the width and height.

Denote the set of N-car positive samples by,

$$D^+_{N\text{-}car} = \{(I_i, B_i^J); |J| = N, B_i^J \subseteq \mathbb{B}_i, i \in [1, n]\}. \quad (5)$$

where all the I_i's have more than N annotated single cars (i.e., $k_i \geq N$). We have,

i) D^+_{1-car} consists of all the single car bounding boxes which do not overlap the other ones in the same image. For $N \geq 2$, D^+_{N-car} is generated iteratively.

ii) In generating D^+_{2-car} (see Fig.4 (a)), for each positive image $(I_i, \mathbb{B}_i) \in D^+$ with $k_i \geq 2$, we enumerate all valid 2-car configurations starting from $B_i^1 \in \mathbb{B}_i$: we first select the current B_i^j as the first car ($1 \leq j \leq k_i$), obtain all the surrounding car bounding boxes $\mathcal{N}_{B_i^j}$ which overlap B_i^j, and then select the second car $B_i^k \in \mathcal{N}_{B_i^j}$ which has the largest overlap if $\mathcal{N}_{B_i^j} \neq \emptyset$ and $(I_i, B_i^J) \notin D^+_{2-car}$ ($J = \{j, k\}$).

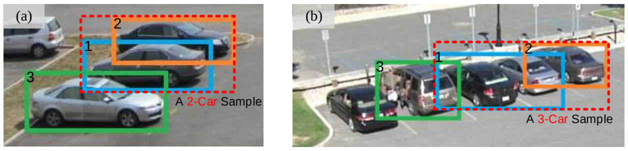


Fig. 4. Illustration of generating multi-car positive samples.

iii) In generating D^+_{N-car} ($N > 2$, see Fig.4 (b)), for each positive image with $k_i \geq N$ and $\exists (I_i, B_i^K) \in D^+_{(N-1)-car}$, we first select the current B_i^K as the seed, obtain the neighbors $\mathcal{N}_{B_i^K}$ each of which overlaps at least one bounding box in B_i^K, and then select the bounding box $B_i^j \in \mathcal{N}_{B_i^K}$ which has the largest overlap and add (I_i, B_i^J) to D^+_{N-car} ($J = K \cup \{j\}$).

Negative Samples. We collect negative samples in images without cars appearing provided in the benchmark datasets and apply the hard negative mining approach during learning parameters as done in the DPM [17].

4.2 Mining Multi-car Contextual Patterns

This section presents the method of learning multi-car patterns in Layer $0-2$ in Fig.3. Considering $N \geq 2$, we use the relative positions of single cars to describe the layout of a multi-car sample $(I_i, B_i^J) \in D^+_{N-car}$. Denote by (cx, cy) the center of a car bounding box ($J = \{1, \cdots, N\}$). Let w_J and h_J be the width and height of the union bounding box of B_i^J respectively. With the center of the first car being the centroid, we define the layout feature by,

$$[\frac{cx_i^2 - cx_i^1}{w_J}, \frac{cy_i^2 - cy_i^1}{h_J}, \cdots, \frac{cx_i^N - cx_i^1}{w_J}, \frac{cy_i^N - cy_i^1}{h_J}]. \quad (6)$$

We cluster these layout features over D^+_{N-car} to get T clusters using k-means. *The obtained clusters are used to specify the And-nodes at Layer 1* in Fig.3. The number of cluster T is specified empirically for different training datasets in our experiments.

In Fig. 5 (top), we visualize the clustering results for D^+_{2-car} on the KITTI [1] and the Parking Lot datasets. Each set of color points represents a 2-car context pattern. In the KITTI dataset, we can observe there are some car-to-car "peak" modes in the dataset (similar to the analyses in [14]), while the context patterns are more diverse in the Parking Lot dataset.

4.3 Mining Occlusion Configurations

In this section we present the method of learning occlusion configurations for single cars in Layer 3 and 4 in Fig.3. We learn the occlusion configurations automatically from a large number of occlusion configurations generated by CAD simulations. Note that the synthetic data are used to learn the occlusion configurations, while the appearance and geometry parameters are still learned from real data.

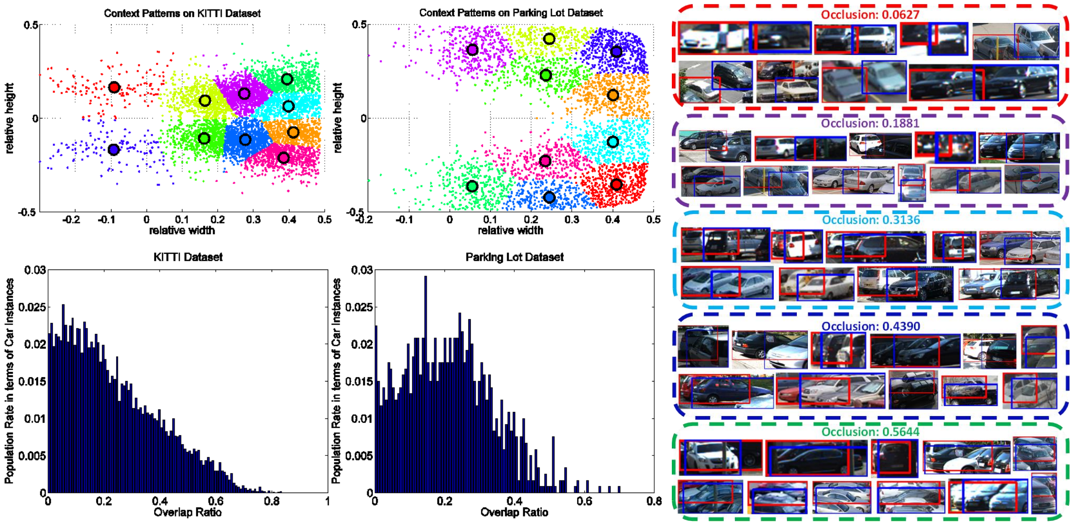


Fig. 5. *Left-Top*: 2-car context patterns on the KITTI dataset [1] and self-collected Parking Lot dataset. Each context pattern is represented by a specific color set, and each circle stands for the center of each cluster. *Left-Bottom*: Overlap ratio histograms of the KITTI dataset and the Parking Lot dataset (we show the occluded cases only). *Right*: some cropped examples with different occlusions. The 2 bounding boxes in a car pair are shown in red and blue respectively. (Best viewed in color).

4.3.1 Generating Occlusion Configurations

As mentioned in Sec.1.2.1, we choose to put 3 cars in generating occlusion configurations. Specifically, we choose the center and 2 other randomly selected positions on a 3×3 grid, and put cars around these grid points to simulate occlusions. See some examples in Fig.2.

The occlusion configurations reflect the four factors: *car type* t, *orientation* ρ , *relative position* r and *camera view* Π. To generate an occlusion configuration, we randomly assign values for these factors, where for each car with type i, $\rho_i \in \{\text{frontal,rear}\}$, $r_i = r_i^{(0)} + \mathrm{d}r$, where $r_i^{(0)}$ is the nominated position for the $i\text{-}th$ car on the 3×3 grid, and $\mathrm{d}r = (\mathrm{d}x, \mathrm{d}y)$ is the relative distance (along x axis and y axis) between sampled position and nominated position of the $i\text{-}th$ car. The camera view is in the range of $\text{azimuth} \in [0, 2\pi]$ and $\text{elevation} \in [0, \pi/4]$, we discretize the view space into B view bins uniformly along the azimuth angle. In the synthesized configurations, a part is treated as occluded if 60% of its area is not visible.

4.3.2 Constructing the Initial And-Or model of Single Cars

With the part-level visibility information, we compute two vectors for each occlusion configuration: The first is a (17 parts$\times B$ camera views) dimension binary valued vector $\vec{v}$ for the visibilities of parts; and the second is a real valued ((1 root +17 parts) $\times B$ camera views$\times 4$) dimension vector $\vec{b}$ for the bounding boxes and parts. In both vectors, entries corresponding to invisible parts are set to 0.

Denoting M as the dimension of the vector $vecv$, and by stacking $vecv$ for N occlusion configurations, we can get an $N \times M$ occlusion matrix $\mathcal{D}$, where the first few rows of this matrix for $B = 8$ is shown in the right side in Fig.6. Note that we have partitioned the view space into B views, so for each row, the visible parts always concentrate in a segment of the vector representing that view.

In learning an initial And-Or model, each row in $\mathcal{D}$ corresponds to a small subtree of the root OR node. In particular, each subtree consists of an And-node as the root and a set of terminal nodes as its children. An example of the data matrix and corresponding initial And-Or model is shown in the middle in Fig.6.

4.3.3 Refining the And-Or Structure

The initial And-Or model is large and redundant, since it has many duplicated occlusion configurations (i.e. duplicated rows in $\mathcal{D}$) and a combinatorial number of part compositions. In the following, we will pursue a compact And-Or structure. The problem can be formulated as:

$$\min \sum_{i}^{N} \mid v_i - v_i(\mathcal{G}) \mid_2^2 + \lambda \mid \mathcal{G} \mid \qquad (7)$$

where v_i is the $i\text{-}th$ row of the data matrix $\mathcal{D}$, $v(\mathcal{G})$ returns its most approximate occlusion configuration generated by the And-Or graph (AOG), $|\mathcal{G}|$ is the number of nodes and edges in the structure, and λ is the trade-off parameter balancing the model precision and complexity. In each view, we assume the number of occlusion branches is not greater than $K (= 4)$.

We solve Eqn.7 using a modified graph compression algorithm similar to [56]. As illustrated in the right side in Fig.6, the algorithm starts from the initial And-Or model, and iteratively combines branches if the introduced loss was smaller than the decrements in complexity term $\lambda|\mathcal{G}|$. This process is equivalent to iteratively finding large blocks of 1s on the corresponding data matrix through row and column permutations, where an example is shown in the bottom in Fig.6. As there are consistently visible parts for each view, the algorithm will quickly converge to the structure shown in Fig.3.

With the refined And-Or model, we compute occlusion configurations (i.e., the consistently visible parts and optional occluded parts) in each view. In addition, the bounding box size and nominal position of each Terminal-node w.r.t. its parent And-node can also be estimated by geometric means of corresponding values in the vector $\vec{b}$. These information will be used to initialize the latent variables of our model in learning the parameters.

Variants of And-Or Models. We will test our model using two types of specifications to be consistent with our two previous conference papers, one is called *And-Or Structure* [6] for occlusion modeling based on CAD simulation without multi-car context components, and the other called *Hierarchical And-Or Model* [7] for occlusion and context. We also compare two methods of part selection in hierarchical

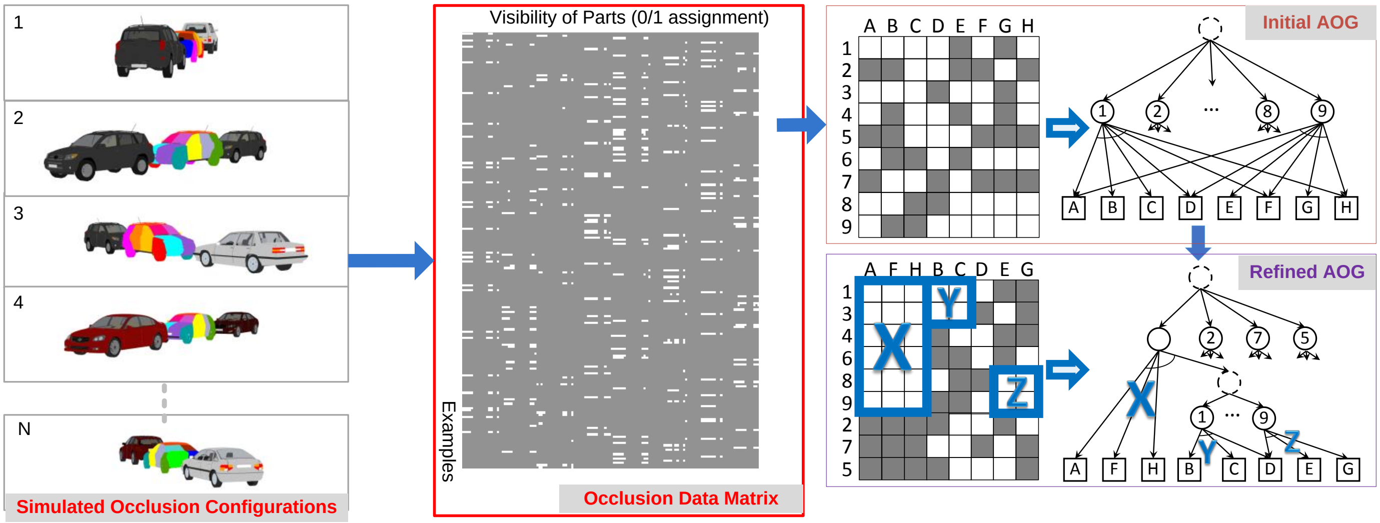


Fig. 6. Illustration of learning occlusion configurations. It consists of three components: (i) Generating occlusion configurations using CAD simulations with 17 semantic parts in total; (ii) Learning the initial And-Or structure based on the data matrix constructed from the simulated occlusion configurations. Each row of the data matrix represents an example and the columns represent the visibility of the 17 semantic parts (a white/gray entry denotes a part is visible/invisible. Each example is represented by an And-node as one child of the root Or-node; (iii) Refining the initial And-Or structure using graph compression algorithm [56] to seek the consistently visible parts (e.g., X) and optional part clusters (e.g., Y and Z).

And-Or model, one is based on the greedy parts as done in the DPM [17], denoted by *AOG+Greedy*, and the other based on the proposed CAD simulation, denoted by *AOG+CAD*.

5 Learning Parameters

With the learned And-Or structure, we adopt the WLSSVM method [15] in learning the parameters $\Theta = (\Theta^{app}, \Theta^{def}, \Theta^{bias})$ (for appearance, deformation and bias). When the occlusion configurations are mined by CAD simulations (i.e., for the two model specifications, And-Or Structure and AOG+CAD), we will use both the *Step 0* and *Step 1* below in learning parameters, otherwise we use *Step 1* only (i.e., for AOG+Greedy).

Step 0: Initializing Parameters with Synthetic Training Data. We learn the initial parameters Θ with synthetic training data (see Fig.10). We randomly superimpose the synthetic positive samples on some randomly selected real images without cars appearing (instead of using white background directly, see Fig.10) to reduce the appearance gap between the synthetic samples and real car samples. In the synthetic data, the parse tree pt for each multi-car positive sample is known except that the positions of parts are allowed to deform.

Step 1: Learning Parameters with Real Training Data. In the real training data, we only have annotated bounding boxes for single cars. The parse tree pt for each multi-car positive sample is hidden except for the multi-car configuration which can be computed based on the annotated bounding boxes of single cars as stated in Sec.4.2. Then, we initialize the parse tree for each positive sample either based on the initial parameters learned in step 0 (for the And-Or structure and AOG+CAD) or using a similar idea as done in learning the mixture of DPMs [17] to initialize the single-car And-nodes for AOG+Greedy. After the initialization, the parameters Θ are learned iteratively under the WLSSVM framework. During learning, we run the DP inference to assign the optimal parse trees for multi-car positive samples.

The objective function to be minimized is defined by,

$$\mathcal{E}(\Theta) = \frac{1}{2}\|\Theta\|^2 + C\sum_{i=1}^{M} L'(\Theta, x_i, y_i) \tag{8}$$

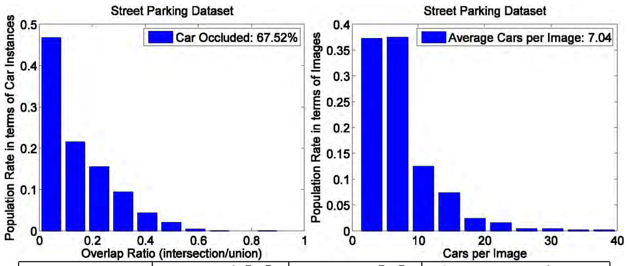
where $x_i \in D^{+}_{N-car}$ represents a training sample ($N \geq 1$) and y_i is the N bounding box(es). $L'(\Theta, x, y)$ is the surrogate loss function,

$$L'(\Theta, x, y) = \max_{pt \in \Omega_{\mathcal{G}}} [score(x, pt; \Theta) + L_{margin}(y, box(pt))] - \max_{pt \in \Omega_{\mathcal{G}}} [score(x, pt; \Theta) - L_{output}(y, box(pt))] \tag{9}$$

where $\Omega_{\mathcal{G}}$ is the space of all parse trees derived from the And-Or model $\mathcal{G}$, $score(x, pt; \Theta)$ computes the score of a parse tree as stated in Sec.3, and $box(pt)$ the predicted bounding box(es) base on the parse tree. As pointed out in [15], the loss $L_{margin}(y, box(pt))$ encourages high-loss outputs to "pop out" of the first term in the RHS, so that their scores get pushed down. The loss $L_{output}(y, box(pt))$ suppresses high-loss outputs in the second term in the right hand side, so the score of a low-loss prediction gets pulled up. More details are referred to [15], [16]. In general, since L' in Eqn.(9) is not convex, the objective function, Eqn.(8) leads to a nonconvex optimization problem. The WLSSVM adopts the CCCP procedure [57] in optimization, which can find a local optima of the objective. The loss function is defined by,

$$L_{\ell,\tau}(y, box(pt)) = \begin{cases} \ell & \text{if } y = \perp \text{ and pt} \neq \perp \\ 0 & \text{if } y = \perp \text{ and pt} = \perp \\ \ell & \text{if } y \neq \perp \text{ and } \exists\ B \in y \\ & \quad \text{with } ov(B, B') < \tau, \forall B' \in box(pt) \\ 0 & \text{if } y \neq \perp \text{ and } ov(B, B') \geq \tau, \\ & \quad \forall\ B \in y \text{ and } \exists B' \in box(pt) \end{cases}, \tag{10}$$

where $\perp$ represents background output and $ov(\cdot,\cdot)$ is the intersection-union ratio of two bounding boxes. Following the PASCAL VOC protocol we have $L_{margin} = L_{1,0.5}$ and $L_{output} = L_{\infty,0.7}$. In practice, we modify the implementation in [18] for our loss formulation.



	Pascal [2]	KITTI [1]	Street Parking
Avg. cars	1.75	≈ 3	7.04

Fig. 7. Top: The distribution of overlap ratio and cars per image on the Street-Parking dataset. Bottom: Comparison of the average number of cars per image.

6 Experiments

In this section, we evaluate our models on four car detection datasets and three car viewpoint estimation dataset and present detail analyses on different aspects of our models. We first introduce two self-collected car datasets of street-parking cars and parking-lot cars respectively (Sec. 6.1), and then evaluate the detection performance of our models on four datasets (Sec. 6.2): the two self-collected datasets, the KITTI car dataset [1] and the PASCAL VOC2007 car dataset [2]. We further analyze the performance of our model w.r.t. different aspects of our models (Sec. 6.3). The performance of car viewpoint estimation is presented in Sec. 6.4.

Training and Testing Time. In all experiments, we utilize a parallel computing technique to train our model. It takes about 9 hours to train an And-Or Structure model and 16 hours to train a hierarchical And-Or Model due to inferring the assignments of part latent variables on positive training examples and mining hard negatives. For detection, it takes about 2 and 3 seconds to process an image with size of 640×480 pixels for a And-Or structure and a hierarchical And-Or model, respectively.

6.1 Datasets

To test our model on occlusion and context modeling, we collected two car datasets [4].

The Street Parking Car Dataset. There are several datasets featuring a large amount of car images [2], [3], [58], [59], but they are not suitable to evaluating occlusion handling, as the proportion of (moderately or heavily) occluded cars is marginal. The recently proposed KITTI dataset [1] contains occluded cars parked along the streets, but it can not fully evaluate the ability of our model since the car views are rather fixed as the video sequences are captured from a car driving on the road (e.g., no birdeye's view). In addition, the average number of cars on each image is still not large enough (mostly 3 cars, see the statistics in the bottom in Fig. 7). To provide a more challenging occlusion dataset, we collected one emphasizing street parking cars with heavy occlusions, diverse viewpoint changes and much larger number of cars per image (see the last two rows in Fig.9). The dataset consists of 881 images. Fig. 7 shows the bounding box overlapping distribution and average number of cars per image. For the simplicity of annotation, we only label the bounding boxes of single cars in each image. We split the dataset into training and testing sets containing 440 and 441 images, respectively.

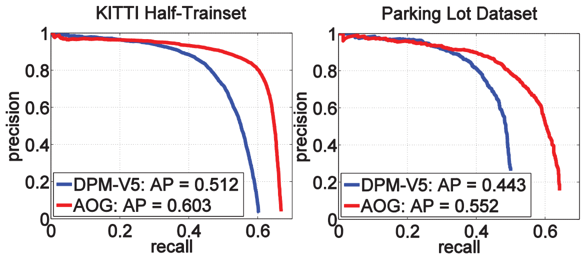


Fig. 8. Precision-recall curves on the test subset splitted from the KITTI trainset (Left) and the Parking Lot dataset (Right).

The Parking Lot Dataset. Our Street Parking Car Dataset provides more viewpoints, however, the context and occlusion configurations are relatively restricted (most cars just compose the head-to-head occlusions). To thoroughly evaluate our models in terms of both context and occlusions, we collected the parking lot car dataset, which has larger occlusion variations and larger number of cars in each image (see the 4-th and 5-th rows in Fig. 9). It contains 65 training images and 63 testing images. Although the number of images is small, the number of cars is noticeably large, with 3,346 cars (including left-right mirrored ones) for training and 2,015 cars for testing.

6.2 Detection

We test our hierarchical And-Or Model on four challenging datasets.

6.2.1 *Results on the KITTI Dataset*

The KITTI dataset [1] contains 7,481 training images and 7,518 testing images, which are captured from an autonomous driving platform. We follow the provided benchmark protocol for evaluation. Since the authors of [1] have not released the test annotations, we test our model in the following two settings.

Training and Testing by Splitting the Trainset. We randomly split the KITTI trainset into the training and testing subsets equally.

Baseline Methods. Since DPM [17] is a very competitive model with source code publicly available, we compare our model with the latest version of DPM (i.e., voc-release5 [18]). The number of components are set to 16 as the baseline methods trained in [1], other parameters are set as default.

Parameter Settings. We consider multi-car contextual patterns with the number of cars $N = 1, 2$. We set the number of context patterns and occlusion configurations to be 10 and 16, respectively. As a result, the learned hierarchical And-Or model has 10 2-car configurations in layer 1, and 16 single car branches in layer 3 (see Fig. 3).

Detection Results. The left figure in Fig. 8 shows the precision-recall curves of DPM and our model. Our model outperforms DPM by 9.1% in terms of average precision (AP). The performance gain comes from both precision and recall, which shows the importance of context and occlusion modeling.

[4] http://www.stat.ucla.edu/~boli/publication/street-parking-release.zip and parking_lot_release.zip

Methods	Easy	Moderate	Hard
mBow [19]	36.02%	23.76%	18.44%
LSVM-MDPM-us [17]	66.53%	55.42%	41.04%
LSVM-MDPM-sv [17], [20]	68.02%	56.48%	44.18%
MDPM-un-BB [17]	71.19%	62.16%	48.43%
OC-DPM [14]	74.94%	65.95%	53.86%
DPM [18] (trained by us)	77.24%	56.02%	43.14%
MV-RGBD-RF [60]	76.40%	69.92%	57.47%
SubCat [44]	84.14%	75.46%	59.71%
3DVP [45]	**87.46%**	75.77%	**65.38%**
Regionlets [61]	84.75%	**76.45%**	59.70%
AOG+Greedy-Half	84.36%	71.88%	59.27%
AOG+Greedy-Full	84.80%	75.94%	60.70%

TABLE 1
Performance comparison (in AP) on the KITTI benchmark [1].

	DPM [18]	And-Or Structure [6]	AOG+Greedy	AOG+CAD
AP	52.0%	57.8%	62.1%	**65.3%**

TABLE 2
Performance comparison (in AP) on the Street Parking dataset [6].

Testing on the KITTI Benchmark. We evaluate our model with two different training data settings: one trained using half training set on the KITTI testset, denoted by AOG+Greedy-Half, and the other trained with full training set, denoted by AOG+Greedy-Full (which has 16 context patterns and 32 occlusion configurations).

The benchmark has three subsets (*Easy, Moderate, Hard*) w.r.t the difficulty of object size, occlusion and truncation. All methods are ranked based on performance in the moderately difficult subset. Our entry in the benchmark is "AOG". Table 1 shows the detection results of our model and other state-of-the-art models. Here, we omit CNN based methods, as they are all anonymous submissions. Details of the benchmark results are available at *http://www.cvlibs.net/datasets/kitti/eval_object.php*.

Our AOG+Greedy-Full outperforms all the DPM-based models. Compared with their best model, OC-DPM [14], our model improved performance on the three subsets by 9.86%, 9.99%, and 6.84% respectively. We also compare with the baseline DPM trained by ourselves using the voc-release5 code [18], and obtain 7.56, 19.92% and 17.56% performance gains on the three stubsets. For other DPM based methods trained by the benchmark authors, our model outperforms the best one - MDPM-un-BB by 13.61%, 13.78% and 12.27% respectively.

Our model is comparable with SubCat [44], 3DVP [45] and Regionlets [61]. We achieve slightly better performance than Regionlets [61] on the *Easy* and *Hard* sets, but lose a bit AP on the *Moderate* set. Though our method obtains better rank than 3DVP [45] on the moderately difficult set, it performs slightly worse on the easy and hard subsets, which shows the promise of 3D occlusion modeling and subcategory clustering [44], [45].

Comparing AOG+Greedy-Half and AOG+Greedy-Full, we can observe that the major improvement (4.06%) of AOG+Greedy-Full comes from the *Moderate* set, while on the *Easy* and *Hard* sets, we obtain small improvement (0.44% and 1.43%, respectively). These results meet some analyses in [62], which indicate there are still large potential improvement on object representation, and much effort should be devoted to improving our current hierarchical And-Or model.

The first 3 rows in Fig. 9 show the qualitative results of our model. The red bounding boxes show successful detection, the blue ones missing detection, and the green ones false alarms. In experiments, our model is robust to detect cars with heavy car-to-car occlusions and background clutters. The failure cases are mainly due to extreme occlusions, extremly low resolution, large car deformation and/or inaccurate (or multiple) bounding box localization.

6.2.2 Results on the Parking Lot Dataset

Evaluation Protocol. We follow the PASCAL VOC evaluation protocol [2] with the overlap of intersection over union being greater than or equal to 60% (instead of original 50%). In practice, we set this threshold to make a compromise between localization accuracy and detection difficulty. The detected cars with bounding box height smaller than 25 pixels do not count as false positives as done in [1]. We compare with the latest version of DPM implementation [18] and set the number of contextual patterns and occlusion configurations to be 10 and 18 respectively.

Detection Results. The right side in Fig. 8 shows the performance comparisons between our model and DPM. Our model obtains 55.2% in AP, which outperforms the latest version of DPM by 10.9%. The fourth and fifth rows in Fig. 9 show the qualitative results. Our model is capable of detecting cars with different occlusions and viewpoints.

6.2.3 Results on the Street Parking Dataset

To compare with the benchmark methods, we follow the evaluation protocol provided in [6].

Results of our model and other benchmark methods are shown in Table 2, our hierarchical And-Or model outperforms DPM [18] and our previous And-Or Structure [6] by 10.1% and 4.3% respectively. We think the performance is improved due to the joint representation of context patterns and occlusion configurations. The last two rows in Fig. 9 show some qualitative examples. Our model is capable of detecting occluded street-parking cars, meanwhile it also has a few inaccurate detection results and misses some cars (mainly due to low resolution).

6.3 Diagnosing the Performance of our Model

In this section, we evaluate various aspects to diagnose the effects of each individual component in our model.

6.3.1 The Effect of Occlusion Modeling

Our And-Or Structure model is based on CAD simulation. Thus in the first analysis, we test the effectiveness of the learned And-Or structure in representing different occlusion configurations. To this purpose, we generate a synthetic dataset using 5,040 3-car synthetic images as our training data, and a mixture of 3,000 3-car and 7-car (placed in a 1 × 7 grid) synthetic images as our testing data. For each generated image, we add the background from the category *None* of the TU Graz-02 dataset [63] and apply Gaussian blur to reduce the boundary effects. Samples of the training and testing data are shown on the left and middle in Fig.10. In experimental comparisons, the best DPM has 16 components and the best And-Or structure has 8 views with

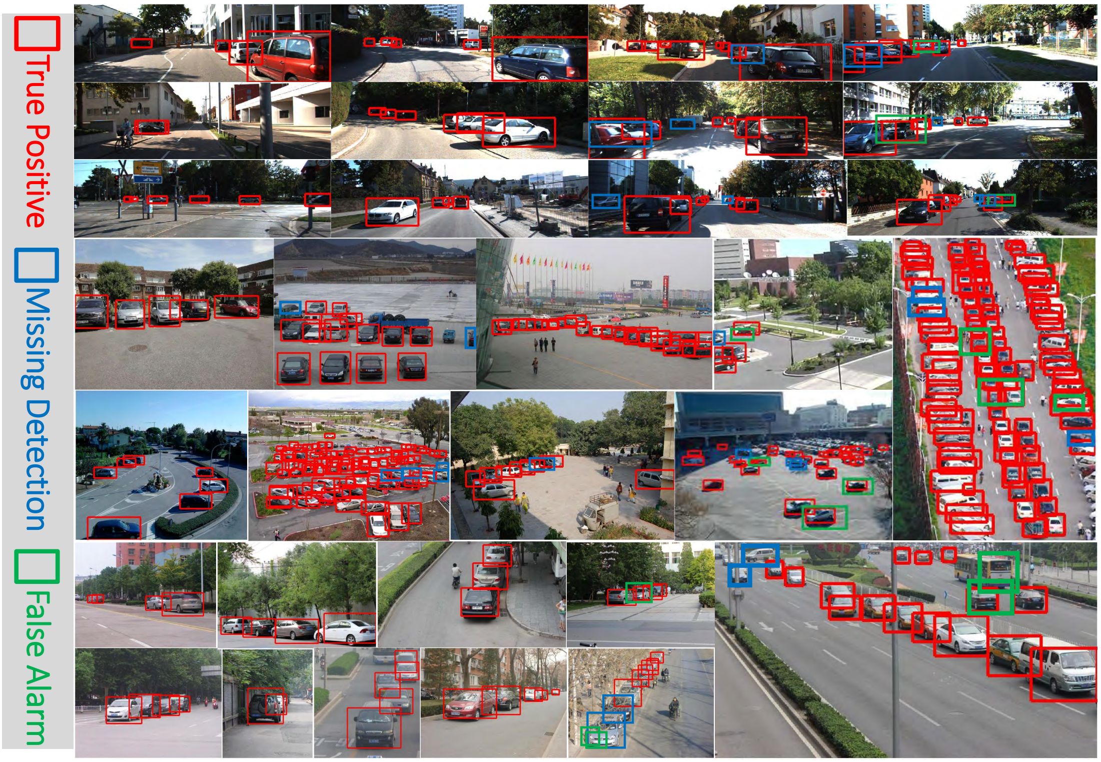


Fig. 9. Examples of successful and failure cases by our model on the KITTI dataset (first 3 rows), the Parking Lot dataset (the 4-th and 5-th rows) and the Street Parking dataset (the last two rows). Best viewed in color and magnification.

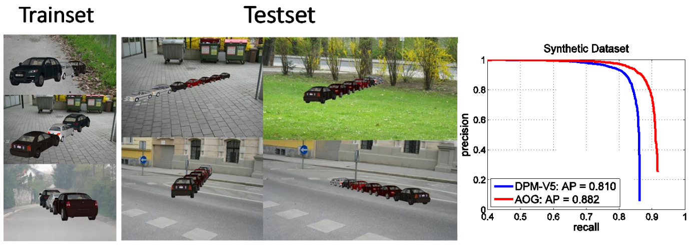


Fig. 10. Left and Middle: Training and testing samples from the synthetic dataset. Right: detection results of DPM and And-Or Structure.

19 occlusion configurations, 5 layers and 111 nodes in total. As shown in the right side in Fig.10, our model outperforms the DPM by 7.2% in AP.

6.3.2 The Effect of CAD Simulation in Real Situations

To verify the effectiveness of our And-Or Structure model in terms of occlusion modeling, we compare it with state-of-the-art DPM [17]. Both of these two models are based on part-level occlusion modeling. The And-Or Structure learns semantic visible parts based on CAD simulations. The DPM handles occlusion implicitly by introducing a trunction feature at each HOG cell. The second and third column in Table 2 show their performance on Street Parking dataset. We can see the semantic visible parts learned from CAD simulations can generalize to real datasets. By adding context, we are interested in whether it affects the effectiveness of occlusion modeling. To compare AOG+Greedy and AOG+CAD fairly, they have the same number of context patterns and occlusion configurations, 8 and 16 respectively. As shown in the fourth and fifth column in Table 2, AOG+CAD performs better than AOG+Greedy, which shows the advantage of modeling occlusion using semantic visible parts.

Fig. 11 shows the inferred part bounding boxes by AOG+Greedy and AOG+CAD. We can observe that the semantic parts in AOG+CAD are meaningful, although they may be not accurate enough in some examples.

car	DPM [18]	And-Or Structure [6]	AOG+Greedy
AP	58.2%	58.7%	**60.6%**

TABLE 3
Performance comparison (in AP) on the PASCAL VOC 2007 [2].

6.3.3 The Effect of Multi-car Context Modeling

The state-of-the-art models are mainly based on single car modeling. To evaluate the effectiveness of context, we compare our hierarchical And-Or model with other non-context models in Table 1. We can see that our model outperforms all other models in different occlusion settings. Specifically, our model outperforms DPM by a large margin (above 10% in AP) on the "Moderate" and "Hard" KITTI test data, which shows context is very important to object detection especially in heavily occluded car-to-car situations.

On the Street Parking dataset, we observe the same results. In Table 2, both AOG+Greedy and AOG+CAD outperform DPM and And-Or Structure by a large margin. Here, AOG+Greedy and AOG+CAD jointly model context and occlusions, while DPM and And-Or Structure model occlusions only.

6.3.4 Performance on General Occlusion Settings

Our model is generalizable in terms of context and occlusion modeling, it can cope with both occlusion and non-occlusion situations. To verify our model on less occluded settings, we use the PASCAL VOC 2007 Car dataset as a testbed. As analyzed by Hoiem, et. al. in [5], cars in the PASCAL VOC dataset do not have much occlusions and car-to-car context.

We first show that our And-Or Structure is capable to detect cars on the PASCAL VOC 2007 as well as the DPM method [18]. To approximate the occlusion configurations

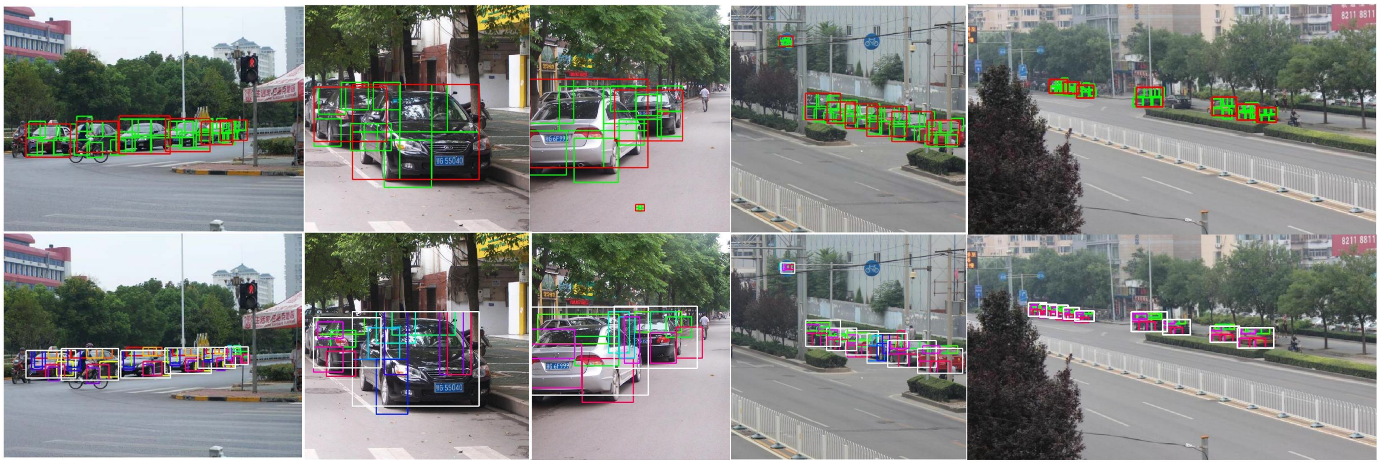


Fig. 11. Visualization of part layouts output by our AOG+Greedy (Top) and AOG+CAD (Bottom). Best viewed in color and magnification.

Pascal VOC 2006 Car Dataset [2]					
	DPM	[64]	[65]	[66]	ours
MPPE	0.69	0.73	**0.86**	0.57	0.73

3D Car Dataset [3]							
	DPM	[64]	[67]	[68]	[30]^1	[30]^2	ours
AP	99.6	96	76.7	99.2	**99.9**	99.7	**99.9**
MPPE	86.3	89	70	85.3	**97.9**	96.3	94

TABLE 4
View Estimation on Pascal VOC 2006 Car Dataset [2] and 3D Car Dataset [3]. [30]^1 and [30]^2 refer to DPM-VOC+VP and DPM-3D-Constraints, respectively.

observed on this dataset, we generate synthetic images with car-to-car occlusions and car self-occlusions. For the car-to-car occlusions, we use the full 3×3 grid instead of the special case in the street parking dataset. Correspondingly, the learned And-Or structure contains branches for self-occlusions as well as those for car-to-car occlusions. On this dataset, the DPM has 6 components and the And-Or structure has 6 views with 10 occlusion configurations, 5 layers and 109 nodes.

The third column in Table 3 shows the performance of our And-Or structure model and the DPM. Our model achieves slightly better recall than DPM, which meets the analysis in [5]. This experiment shows that our And-Or structure method does not lose performance in general datasets.

Then, we verify our hierarchical And-Or model is capable to detect cars on the PASCAL VOC 2007 as well as other single object models. We compare with the latest version of DPM [18]. The APs are 60.6% (our model) and 58.2% (DPM) respectively (Table 3).

6.4 View Estimation

With the help of CAD simulations, our And-Or Structure model can compute the viewpoints of detected cars. To verify the capability of view estimation, we perform 2 experiments.

Firstly, we report the mean precision in pose estimation (MPPE), equivalent to the means of confusion matrix diagonals, on both the Pascal VOC 2006 car dataset [69] and the 3D Object dataset [3]. The 3D Object Classes dataset [3] is introduced in 2007. For each class, it has images of 10 different object instances with 8 different poses. We follow the evaluation protocol described in [3]: 7 randomly selected car instances are used for training, and 3 instances for testing. The 2D car bounding boxes are computed from the annotated segmentation masks. The negative examples are collected from the PASCAL VOC 2007 car dataset. For the VOC 2006 car database [69], there are 469 cars with viewpoint labels (frontal, rear, left and right). We only use these labeled images with the standard training/test split. The detection performance is evaluated through precision-recall (PR) curve. For view estimation, the two datasets emphasize visible cars. Our And-Or structure has 8 views with 8 (self-occlusion) branches, 5 layers and 90 nodes. Table 4 shows the comparison of our model with the state-of-the-art methods on these two datasets. Our model is comparable to or better than some recently proposed models [30], [64], [65].

Secondly, we compare our model with the state-of-the-art models on the recently proposed PASCAL3D+ Dataset [4]. This dataset augments 12 rigid categories in the PASCAL VOC 2012 [2] with 3D annotations by fitting CAD models with 2D images semi-manually. It is a challenging dataset for 3D object detection and pose estimation. We test on the car category. We use the metric - Average Viewpoint Precision (AVP) [4] to simultaneously evaluate 2D bounding box localization and viewpoint estimation. In computing the AVP, a candidate detection is considered to be a true positive if and only if the bounding box overlap is larger than 50% and the viewpoint is correct.

Table 5 shows the results of our model and the state-of-the-art methods. Our method is better than VDPM [4] and a deep-cnn-feature-based model (decaf) [21]. Our And-Or Structure is comparable with [30], which also used CAD models to learn viewpoints and part-level car geometry.

7 Conclusion

In this paper, we present an And-Or model to represent context and occlusion for car detection and viewpoint estimation. The model structure is learned by mining multi-car contextual patterns and occlusion configurations at three

	VDPM [4]	DPM-VOC+VP [30]	(fisher+spm) [21]	(decaf) [21]	our And-Or Structure
4 views	37.2%/20.2%	45.6%/36.9%	36.1%/28.9%	36.1%/24.1%	43.0%/34.3%
8 views	37.3%/23.5%	47.6%/36.6%	36.1%/26.6%	36.1%/23.3%	44.9%/33.2%
16 views	36.6%/18.1%	46.0%/29.6%	36.1%/19.6%	36.1%/19.4%	43.2%/27.6%
24 views	36.3%/13.7%	42.1%/24.6%	36.1%/15.9%	36.1%/16.7%	41.1%/22.9%

TABLE 5
The results of VDPM, DPM-VOC+VP and And-Or Structure on the PASCAL3D+ Car Dataset [4]. The first number indicates the average precision (AP) for detection and the second number shows the average viewpoint precision (AVP) for joint object detection and view estimation.

levels: a) multi-car layouts, b) single car and c) parts. Our model is organized in a directed and acyclic graph structure so the efficient DP algorithm can be used in inference. The model parameters are learned by WLSSVM [15]. Experimental results show that our model is effective in modeling context and occlusion information in complex situations, and achieves better performance over state-of-the-art car detection methods and comparable performance on viewpoint estimation.

There are two main limitations in our current implementation. The first one is that we exploited the multi-car contextual patterns using 2-car composite only. In the scenarios similar to street parking cars and parking lot cars, we could explore multi-car context with more than 2 spatially-aligned cars, as well as 3D scene parsing context [70]. The second one is that we utilized only the HOG features for appearance. Based on the recent progress on feature learning by convolutional neural network (CNN) [71], [72], we can also substitute the HOG by the CNN features. Both aspects are addressed in our on-going work and may potentially improve the performance.

Meanwhile, we are applying the proposed method to other object categories and studying different ways of mining contextual patterns and occlusion configurations (e.g., integrating with the And-Or quantization methods for 2D object modeling [24] and 3D car modeling [47]).

Acknowledgments

B. Li is supported by China 973 Program under Grant no. 2012CB316300. T.F. Wu and S.C. Zhu are supported by DARPA MSEE project FA 8650-11-1-7149, MURI grant ONR N00014-10-1-0933, and NSF IIS1018751. We thank Dr. Wenze Hu for helpful discussions. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of one GPU.

References

[1] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *CVPR*, 2012.

[2] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.

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

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

[5] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *ECCV*, 2012.

[6] B. Li, W. Hu, T.-F. Wu, and S.-C. Zhu, "Modeling occlusion by discriminative and-or structures," in *ICCV*, 2013.

[7] B. Li, T. Wu, and S.-C. Zhu, "Integrating context and occlusion for car detection by hierarchical and-or model," in *ECCV*, 2014.

[8] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 4, pp. 259–362, Jan. 2006.

[9] P. Felzenszwalb and D. McAllester, "Object detection grammars," University of Chicago, Computer Science TR-2010-02, Tech. Rep., 2010.

[10] M. Sadeghi and A. Farhadi, "Recognition using visual phrases," in *CVPR*, 2011.

[11] B. Li, X. Song, T. Wu, W. Hu, and M. Pei, "Coupling-and-decoupling: A hierarchical model for occlusion-free object detection," *Pattern Recognition*, vol. 47, no. 10, pp. 3254 – 3264, 2014.

[12] S. Tang, M. Andriluka, and B. Schiele, "Detection and tracking of occluded people," in *BMVC*, 2012.

[13] W. Ouyang and X. Wang, "Single-pedestrian detection aided by multi-pedestrian detection," in *CVPR*, 2013.

[14] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *CVPR*, 2013.

[15] R. Girshick, P. Felzenszwalb, and D. McAllester, "Object detection with grammar models," in *NIPS*, 2011.

[16] D. McAllester and J. Keshet, "Generalization bounds and consistency for latent structural probit and ramp loss," in *NIPS*, 2011.

[17] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *TPAMI*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[18] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester, "Discriminatively trained deformable part models, release 5," http://people.cs.uchicago.edu/ rbg/latent-release5/.

[19] J. Behley, V. Steinhage, and A. Cremers, "Laser-based Segment Classification Using a Mixture of Bag-of-Words," in *IROS*, 2013.

[20] A. Geiger, C. Wojek, and R. Urtasun, "Joint 3d estimation of objects and scene layout," in *NIPS*, 2011.

[21] A. Ghodrati, M. Pedersoli, and T. Tuytelaars, "Is 2d information enough for viewpoint estimation?" in *BMVC*, 2014.

[22] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vision*, vol. 57, no. 2, pp. 137–154, May 2004.

[23] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *TPAMI*, vol. 34, no. 4, pp. 743–761, Apr. 2012.

[24] X. Song, T.-F. Wu, Y. Jia, and S.-C. Zhu, "Discriminatively trained and-or tree models for object detection," in *CVPR*, 2013.

[25] K. Grauman and B. Leibe, *Visual Object Recognition*, ser. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2011.

[26] A. Andreopoulos and J. K. Tsotsos, "50 years of object recognition: Directions forward." *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827–891, 2013.

[27] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, and C. Gao, "Object class detection: A survey," *ACM Comput. Surv.*, vol. 46, no. 1, pp. 10:1–10:53, Jul. 2013.

[28] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, "Latent hierarchical structural learning for object detection," in *CVPR*, 2010.

[29] H. Azizpour and I. Laptev, "Object detection using strongly-supervised deformable part models," in *ECCV*, 2012.

[30] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Teaching 3d geometry to deformable part models," in *CVPR*, 2012.

[31] S. Branson, P. Perona, and S. Belongie, "Strong supervision from weak annotation: Interactive training of deformable part models," in *ICCV*, 2011.

[32] B. Wu and R. Nevatia, "Detection and tracking of multiple, partially occluded humans by bayesian combination of edgelet based part detectors," *International Journal of Computer Vision*, vol. 75, no. 2, pp. 247–266, 2007.

[33] X. Wang, T. Han, and S. Yan, "An hog-lbp human detector with partial occlusion handling," in *ICCV*, 2009.

[34] M. Hejrati and D. Ramanan, "Analyzing 3d objects in cluttered images," in *NIPS*, 2012.

[35] X. Burgos-Artizzu, P. Perona, and P. Dollár, "Robust face landmark estimation under occlusion," in *ICCV*, 2013.

[36] C. Desai and D. Ramanan, "Detecting actions, poses, and objects with relational phraselets." in *ECCV*, 2012.

[37] T. Gao, B. Packer, and D. Koller, "A segmentation-aware object detection model with occlusion handling," in *CVPR*, 2011.

[38] G. Duan, H. Ai, and S. Lao, "A structural filter approach to human detection," in *ECCV*, 2010.

[39] M. Mathias, R. Benenson, R. Timofte, and L. Van Gool, "Handling occlusions with franken-classifiers," in *ICCV*, 2013.

[40] X. Yu, Z. Lin, J. Brandt, and D. N. Metaxas, "Consensus of regression for occlusion-robust facial feature localization," in *ECCV*, 2014.

[41] M. Z. Zia, M. Stark, and K. Schindler, "Explicit Occlusion Modeling for 3D Object Class Representations," in *CVPR*, 2013.

[42] G. Ghiasi and C. C. Fowlkes, "Occlusion coherence: Localizing occluded faces with a hierarchical deformable part model," in *CVPR*, 2014.

[43] G. Ghiasi, Y. Yang, D. Ramanan, and C. C. Fowlkes, "Parsing occluded people," in *CVPR*, 2014.

[44] E. Ohn-Bar and M. Trivedi, "Learning to detect vehicles by clustering appearance patterns," *TITS*, 2015.

[45] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3d voxel patterns for object category recognition," in *CVPR*, 2015.

[46] J. Zhu, T. Wu, S.-C. Zhu, X. Yang, and W. Zhang, "Learning reconfigurable scene representation by tangram model," in *WACV*, 2012.

[47] W. Hu and S.-C. Zhu, "Learning 3d object templates by quantizing geometry and appearance spaces," *TPAMI*, vol. 37, no. 6, pp. 1190–1205, 2015.

[48] Y. Yang, S. Baker, A. Kannan, and D. Ramanan, "Recognizing proxemics in personal photos," in *CVPR*, 2012.

[49] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for multi-class object layout," *IJCV*, vol. 95, no. 1, pp. 1–12, 2011.

[50] D. Hoiem, A. Efros, and M. Hebert, "Putting objects in perspective," *IJCV*, vol. 80, no. 1, pp. 3–15, 2008.

[51] Z. Tu and X. Bai, "Auto-context and its application to high-level vision tasks and 3d brain image segmentation," *TPAMI*, vol. 32, no. 10, pp. 1744–1757, Oct. 2010.

[52] G. Chen, Y. Ding, J. Xiao, and T. X. Han, "Detection evolution with multi-order contextual co-occurrence," in *CVPR*, 2013.

[53] K. Matzen and N. Snavely, "Nyc3dcars: A dataset of 3d vehicles in geographic context," in *ICCV*, 2013.

[54] Y. Zhang, S. Song, P. Tan, and J. Xiao, "Panocontext: A whole-room 3d context model for panoramic scene understanding," in *ECCV*, 2014.

[55] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[56] Z. Si and S.-C. Zhu, "Learning and-or templates for object recognition and detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 9, pp. 2189–2205, 2013.

[57] A. L. Yuille and A. Rangarajan, "The Concave-Convex Procedure (CCCP)," in *NIPS*, 2001.

[58] B. Leibe and B. Schiele, "Analyzing appearance and contour based methods for object categorization," in *CVPR*, 2003.

[59] M. Ozuysal, V. Lepetit, and P. Fua, "Pose estimation for category specific multiview object localization," in *CVPR*, 2009.

[60] A. Gonzalez, G. Villalonga, D. V. J. Xu, J. Amores, and A. Lopez, "Multiview random forest of local experts combining rgb and lidar data for pedestrian," in *IEEE Intelligent Vehicles Symposium (IV)*, 2015.

[61] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *ICCV*, December 2013.

[62] X. Zhu, C. Vondrick, D. Ramanan, and C. C. Fowlkes, "Do we need more training data or better models for object detection?" in *BMVC*, 2012.

[63] A. Opelt and A. Pinz, "Object Localization with Boosting and Weak Supervision for Generic Object Recognition," in *SCIA*, 2005.

[64] R. J. Lopez-Sastre, T. T., and S. Savarese, "Deformable part models revisited: A performance evaluation for object category pose estimation," in *ICCV-WS CORP*, 2011.

[65] C. Gu and X. Ren, "Discriminative Mixture-of-Templates for Viewpoint Classification," in *ECCV*, 2010.

[66] M. Sun, H. Su, S. Savarese, and L. Fei-Fei, "A multi-view probabilistic model for 3d object classes," in *CVPR*, 2009.

[67] J. Liebelt and C. Schmid, "Multi-view object class detection with a 3D geometric model," in *CVPR*, 2010.

[68] D. Glasner, M. Galun, S. Alpert, R. Basri, and G. Shakhnarovich, "Viewpoint-aware object detection and pose estimation," in *ICCV*, 2011.

[69] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool, "The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results," http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[70] X. Liu, Y. Zhao, and S. Zhu, "Single-view 3d scene parsing by attributed grammar," in *CVPR*, 2014.

[71] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[72] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, 2014.

Tianfu Wu received a Ph.D. degree in Statistics from University of California, Los Angeles (UCLA) in 2011. He is currently a research assistant professor in the center for vision, cognition, learning and autonomy (VCLA) at UCLA. His research interests include: (i) Statistical learning of large scale hierarchical and compositional models (e.g., And-Or graphs) from images and videos. (ii) Statistical inference by learning near-optimal cost-sensitive decision policies. (iii) Statistical theory of performance guaranteed learning algorithm and inference procedure.

Bo Li received a B.S. degree from Beijing Institute of Technology in 2010. He is currently a Ph.D. student in School of Computer Science and Technology, Beijing Institute of Technology, and a visiting student at Center for Vision, Cognition, Learning and Autonomy (VCLA) at the University of California, Los Angeles (UCLA). His research interests are in pattern recognition, machine learning and computer vision, with a focus on car detection in terms of both 2D and 3D models.

Song-Chun Zhu received a Ph.D. degree from Harvard University in 1996. He is currently a professor of Statistics and Computer Science at UCLA, and the director of the Center for Vision, Cognition, Learning and Autonomy. He has published over 160 papers in computer vision, statistical modeling and learning, cognition, and visual arts. He received a number of honors, including the J.K. Aggarwal prize from the Int'l Association of Pattern Recognition in 2008 for "contributions to a unified foundation for visual pattern conceptualization, modeling, learning, and inference", the David Marr Prize in 2003 with Z. Tu et al. for image parsing, twice Marr Prize honorary nominations in 1999 for texture modeling and in 2007 for object modeling with Z. Si and Y.N. Wu. He received the Sloan Fellowship in 2001, a US NSF Career Award in 2001, and an US ONR Young Investigator Award in 2001. He received the Helmholtz Test-of-time award in ICCV 2013, and he is a Fellow of IEEE since 2011.