Integrating Context and Occlusion for Car Detection by Hierarchical And-Or Model

Bo Li[1,2], Tianfu Wu[2,⋆], and Song-Chun Zhu[2]

[1] Beijing Lab of Intelligent Information Technology, Beijing Institute of Technology
[2] Department of Statistics, University of California, Los Angeles
boli86@bit.edu.cn, {tfwu,sczhu}@stat.ucla.edu

Abstract. This paper presents a method of learning reconfigurable hierarchical And-Or models to integrate context and occlusion for car detection. The And-Or model represents the regularities of car-to-car context and occlusion patterns at three levels: (i) layouts of spatially-coupled N cars, (ii) single cars with different viewpoint-occlusion configurations, and (iii) a small number of parts. The learning process consists of two stages. We first learn the structure of the And-Or model with three components: (a) mining N-car contextual patterns based on layouts of annotated single car bounding boxes, (b) mining the occlusion configurations based on the overlapping statistics between single cars, and (c) learning visible parts based on car 3D CAD simulation or heuristically mining latent car parts. The And-Or model is organized into a directed and acyclic graph which leads to the Dynamic Programming algorithm in inference. In the second stage, we jointly train the model parameters (for appearance, deformation and bias) using Weak-Label Structural SVM. In experiments, we test our model on four car datasets: the KITTI dataset [11], the street parking dataset [19], the PASCAL VOC2007 car dataset [7], and a self-collected parking lot dataset. We compare with state-of-the-art variants of deformable part-based models and other methods. Our model obtains significant improvement consistently on the four datasets.

Keywords: Car Detection, Context, Occlusion, And-Or Graph.

1 Introduction

The recent literature of object detection has been focused on three aspects to improve accuracy performance: using hierarchical models such as discriminatively trained deformable part-based models (DPM) [8] and And-Or tree models [27], modeling occlusion implicitly or explicitly [19,26,28,23,25], and exploiting contextual information [30,6,17,29,4]. In this paper, we present a method of learning reconfigurable hierarchical And-Or models to integrate context and occlusion for car detection in the wild, e.g., car detection in the recently proposed challenging KITTI dataset [11] and the Street-Parking dataset [19].

⋆ Corresponding author.

D. Fleet et al. (Eds.): ECCV 2014, Part VI, LNCS 8694, pp. 652–667, 2014.

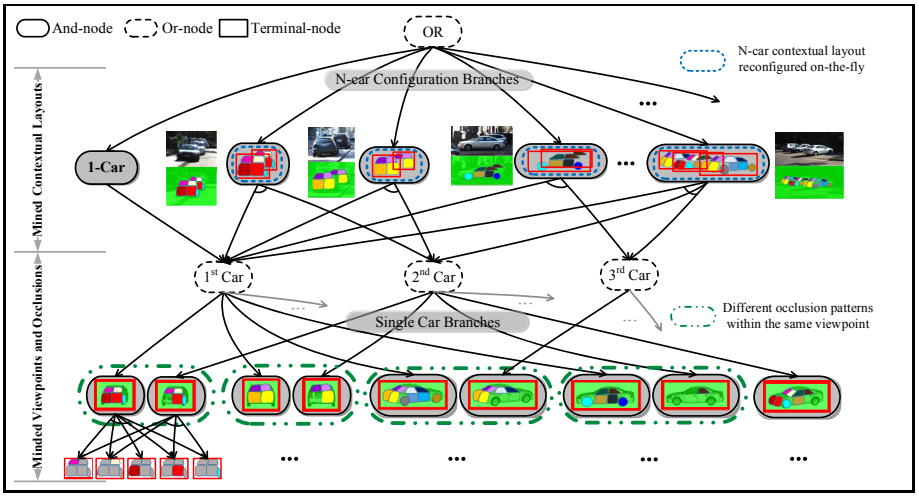

Fig. 1. Illustration of our reconfigurable hierarchical And-Or model for car detection. It represents contextual layouts and viewpoint-occlusion patterns jointly by modeling strong spatially-coupled N-car (e.g., $N = 1, 2, 3$) together and composing visible parts explicitly for single cars. See text for details. (Best viewed in color)

Fig. 1 illustrates the And-Or model learned for car detection. It is organized into a directed and acyclic graph (DAG) and embeds object detection grammar [32,9]. It consists of three types of nodes: And-nodes representing decompositions, Or-nodes representing structural variations and Terminal-nodes grounding symbols (i.e., objects and parts) to image data.

i) The root Or-node represents different N-car configurations which capture both car viewpoints (when $N \geq 1$) and car-to-car contextual information (when $N > 1$). Each configuration is then represented by an And-node (e.g., car pairs and car triples shown in the figure). The contextual information reflects the layout regularities of N cars in real scenarios (such as cars in a parking lot and street-parking cars).
ii) A specific N-car configuration is represented by an And-node which is decomposed into N 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 viewpoints and occlusion patterns (e.g., the car in the back of a car-pair can have different occluding situations due to the layouts).
iii) Each viewpoint-occlusion pattern is represented by an And-node which is further decomposed into parts. Parts are learned using car 3D CAD simulation as done in [19] or the heuristic method as done in DPM [8]. The green dashed bounding boxes show some examples corresponding to different occlusion patterns (i.e., visible parts) within the same viewpoint.

The proposed And-Or model is flexible and reconfigurable to account for the large variations of car-to-car layouts and viewpoint-occlusion patterns in complex

situations. Reconfigurability is one of the most desired property in hierarchical models. In training data, only bounding boxes of single cars are given. We learn the And-Or model with two stages:

i) *Learning the structure of the hierarchical And-Or model.* Both the N-car configurations and viewpoint-occlusion patterns of single cars are mined automatically based on the annotated single car bounding boxes in training data (i.e., weakly-supervised). The learned structure is a DAG since we have both single-car-sharing and part-sharing, which facilitates the Dynamic Programming (DP) algorithm in inference.
ii) *Learning the parameters for appearance, deformation and bias* using Weak-Label Structural SVM (WLSSVM) [13,22]. In our model, we learn appearance templates and deformation models for single cars and parts, and the composed appearance templates for a N-car configuration is inferred on-the-fly (i.e., reconfigurability). So, our model can express a large number of N-car configurations with different compatible viewpoint-occlusion combinations of single cars.

In experiments, we test our model on four car datasets: the KITTI dataset [11], the Street-Parking dataset [19], the PASCAL VOC2007 car dataset [7] and a self-collected Parking Lot dataset (to be released with this paper). Experimental results show that the proposed hierarchical And-Or model is capable of modeling context and occlusion effectively. Our model outperforms different state-of-the-art variants of DPM [8] (including the latest implementation [14]) on all the four datasets, as well as other state-of-the-art models [2,12,25,19] on the KITTI and the Street-Parking datasets. The code and data will be available on the author's homepage[1].

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

2 Related Work and Our Contributions

Single Object Models and Occlusion Modeling. Hierarchical models are widely used in recent literature of object detection and most existing works are devoted to learning a single object model. Many work share the similar spirit to the deformable part-based model [8] (which is a two-layer structure) by exploring deeper hierarchy and global part configurations [27,31,13], with strong manually-annotated parts [1] or available 3D CAD models [24], or by keeping

[1] `http://www.stat.ucla.edu/~tfwu/project/OcclusionModeling.htm`

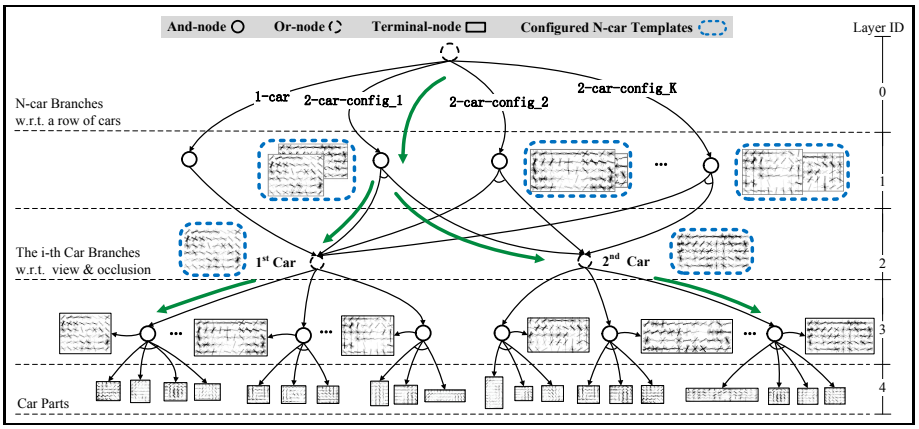

Fig. 2. The learned And-Or model for car detection (only a portion of the whole model is shown here for clarity). The node in layer 0 is the root Or-node, which has a set of child And-nodes representing different N-car configurations in layer 1 ($N \leq 2$ is considered). The nodes in layer 2 represent single car Or-nodes, each of which has a set of child And-nodes representing single cars with different viewpoints and occlusion patterns. We learn appearance templates for single cars and their parts (nodes in layer 3 and 4), and the composite templates for a N-car is reconfigured on-the-fly in inference (as illustrated by the green solid arrows). (Best viewed in color)

human in-the-loop [3]. To address the occlusion problem, methods of regularizing part visibilities are used in learning [15,19]. Those models do not represent contextual information, and usually learn another separate context model using the detection scores as input features. Recently, an And-Or quantization method is proposed to learn And-Or tree models [27] for generic object detection in PASCAL VOC [7] and learn car 3D And-Or models [18] respectively, which could be useful in occlusion modeling.

Object-Pair and Visual Phrase Models. To account for the strong co-occurrence, object-pair [20,28,23,25] and visual phrase [26] methods implicitly model 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 in occluded situations, the object-pair or visual phrase are often manually designed and fixed (i.e., not reconfigurable in inference), and as investigated in the KITTI dataset [25], their performance are worse than original DPM in complex scenarios.

Context Models. Many context models have been exploited in object detection showing performance improvement [30,6,17,29,4]. In [29], Tu and Bai integrate the detector responses with background pixels to determine the foreground pixels. In [4], Chen, et. al. propose a multi-order context representation to take

advantage of the co-occurrence of different objects. Most of them model objects and context separately.

This paper aims to integrate context and occlusion by a hierarchical And-Or model and makes three main contributions to the field of car detection as follows.

i) It proposes a hierarchical And-Or model to integrate context and occlusion patterns. The proposed model is flexible and reconfigurable to account for large structure, viewpoint and occlusion variations.
ii) It presents a simple, yet effective, approach to mine context and occlusion patterns from weakly-labeled training data.
iii) It introduce a new parking lot car dataset, and outperforms state-of-the-art car detection methods in four challenging datasets.

3 The And-Or Model for Car Detection

3.1 The And-Or Model and Scoring Functions

Our And-Or model follows the image grammar framework proposed by Zhu and Mumford [32] which has shown expressive power to represent a large number of configurations using a small dictionary. In this section, we first introduce the notations to define the And-Or model and its scoring function. Fig. 2 shows the learned car And-Or model which has 5 layers.

The 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 set of nodes consisting of three subsets of And-nodes, Or-nodes and Terminal-nodes respectively, E the set of edges organizing all the nodes into a DAG, and $\Theta = (\Theta^{app}, \Theta^{def}, \Theta^{bias})$ the set of parameters (for appearance, deformation and bias respectively, to be defined later). Denote by $ch(v)$ the set of child nodes of a node $v \in \mathcal{V}_{\text{And}} \cup \mathcal{V}_{\text{Or}}$.

Appearance Features. We adopt the Histogram of Oriented Gradients (HOG) feature [5,8] to describe car appearance. Let I be an image defined on a 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 $\mathcal{H}$.

Deformation Features. We allow local deformation when composing the child nodes into a parent node (e.g., composing car parts into a single car or composing two single cars into a car-pair). In our model, car parts are placed at twice the spatial resolution w.r.t. single cars, while single cars and composite N-cars are placed 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 symbol (i.e., a single car or a car part) to image data (see Layer 3 and 4 in Fig.2). 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 $\theta^{def}_{t|A} \subset \Theta^{def}$ the deformation parameters. Given the position $p_A = (l_A, x_A, y_A)$ of parent node A, the scoring function of node t is defined by,

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

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 $(x_t, y_t) = 2^{s_t}(x_A, y_A) + a_{t|A} + \delta$, and $\Phi^{app}(\mathcal{H}, p_t)$ the extracted HOG features. $< \cdot, \cdot >$ denotes the inner product.

An **And-node** $A \in \mathcal{V}_{\text{And}}$ represents a decomposition of a large entity (e.g., a N-car layout at Layer 1 or a single car at Layer 3 in Fig.2) into its constituents (e.g., N single cars or a small number of car parts). The scoring function of node A is defined by,

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

where $b_A \in \Theta^{bias}$ is the bias term. Each single car And-node (at Layer 3) can be treated as the And-Or Structure proposed in [19] or the DPM [8]. So, our model is very flexible to incorporate state-of-the-art single object models. For N-car layout 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 at Layer 0 and the i-th car node at Layer 2 in Fig.2). For the root Or-node O, when placing at the position $p \in \Lambda$, the scoring function is defined by,

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

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

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

where $p_v = (l_v, x_v, y_v)$ with $l_v = l_A$ and $(x_v, y_v) = (x_A, y_A) + \delta$.

3.2 The DP Algorithm in Detection

In detection, we place the And-Or model at all positions $p \in \Lambda$ and retrieve the parse trees for all positions at which the scores are greater than the detection threshold. A *parse tree* is an instantiation of the And-Or model by selecting the best child of each encountering Or-node as illustrated by the green arrows in Fig.2. Thank to the DAG structure of our And-Or model, we can utilize the efficient DP algorithm in detection which consists of two stages:

- Following the depth-first-search (DFS) order of nodes in the And-Or model, the bottom-up pass computes appearance score maps and deformed score maps for the whole feature pyramid $\mathcal{H}$ for all Terminal-nodes, And-nodes and Or-nodes. The deformed score maps can be computed efficiently by the generalized distance transform [10] algorithm as done in [8].
- In the top-down pass, we first find all the positions $\mathbb{P}$ for the root Or-node O with score $score(O, p) \geq \tau, p \in \mathbb{P} \subset \Lambda$. Then, following the breadth-first-search (BFS) order of nodes, we can retrieve the parse tree at each p.

Post-processing. To generate the final detection results of single cars for evaluation, we apply N-car guided non-maximum suppression (NMS), since we deal with occlusion: (i) Overlapped N-car detection candidates might report multiple predictions for the same single car. For example, if a car is shared by two neighboring 2-car detection candidates, it will be reported twice; (ii) Some of the cars in a N-car detection candidate are highly overlapped due to occlusion, and if we directly use conventional NMS we will miss the detection of the occluded cars. In our N-car guided NMS, we enforce that all the N single car bounding boxes in a N-car prediction will not be suppressed by each other. The similar idea is also used in [28].

4 Learning the Model Structure by Mining Context and Viewpoint-Occlusion Patterns

In this section, we present the methods of learning the structure of our And-Or model by mining context and viewpoint-occlusion patterns in the positive training dataset. Denote by $D^+ = \{(I_1, \mathbb{B}_1), \cdots, (I_n, \mathbb{B}_n)\}$ the positive training dataset where $\mathbb{B}_i = \{B_i^j = (x_i^j, y_i^j, w_i^j, h_i^j)\}_{j=1}^{k_i}$ is the set of k_i annotated single car bound boxes in image I_i (where (x, y) is the left-top corner and (w, h) the width and height).

Generating the N-car positive samples from D^+. Denote the set of N-car positive samples by,

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

we have,

- 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.
- To generate D^+_{2-car}, 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 $B_i^1 \in \mathbb{B}_i$: (i) select the current B_i^j as the first car $(1 \leq j \leq k_i)$, (ii) obtain all the surrounding car bounding boxes $\mathcal{N}_{B_i^j}$ which overlap B_i^j, and (iii) 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 $(I_i, B_i^J) \notin D^+_{2-car}$ (where $J = \{j, k\}$).

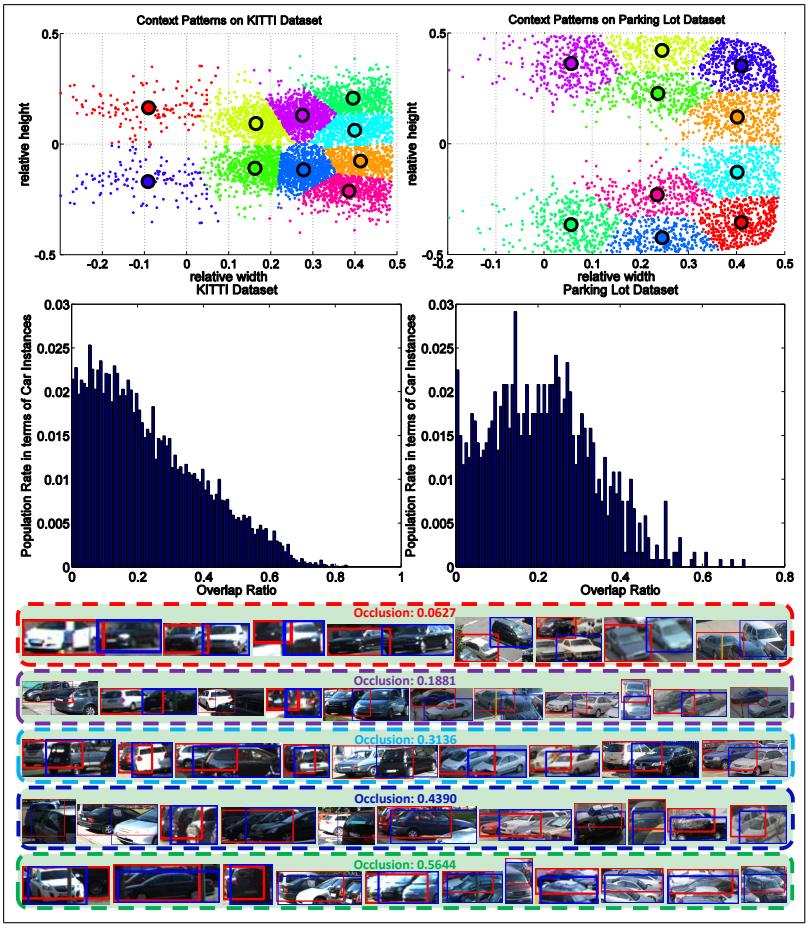

Fig. 3. *Top*: 2-car context patterns on the KITTI dataset [11] and self-collected Parking Lot dataset. Each context pattern is represented by a specific color set, and each circle stands for the center of each cluster. *Middle*: Overlap ratio histograms of the KITTI dataset and the Parking Lot dataset (we show the occluded cases only). *Bottom*: 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).

– To generate D^+_{N-car} $(N > 2)$, for each positive image with $k_i \geq N$ and $\exists(I_i, B_i^K) \in D^+_{(N-1)-car}$, (i) select the current B_i^K as the seed, (ii) obtain the neighbors $\mathcal{N}_{B_i^K}$ each of which overlap at least one bounding box in B_i^K, (iii) 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} (where $J = K \cup \{j\}$) if valid.

4.1 Mining *N*-car Context Patterns

Consider $N \geq 2$, we use the relative positions of single cars to describe the layout of a N-car sample $(I_i, B_i^J) \in D^+_{N-car}$. Denote by (cx, cy) the center of a car bounding box. Assume $J = \{1, \cdots, N\}$. Let w_J and h_J be the width and height of the union bounding box of B_i^J. With the center of the first car being the centroid, we define the layout feature by $[\frac{cx_i^2 - cx_i^1}{w_J}, \frac{cy_i^2 - cy_i^1}{h_J}, \cdots, \frac{cx_i^N - cx_i^1}{w_J}, \frac{cy_i^N - cy_i^1}{h_J}]$. 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.2. The number of cluster T is specified empirically for different training datasets in our experiments.

In Fig. 3 (top), we visualize the clustering results for D^+_{2-car} on the KITTI [11] and self-collected Parking Lot datasets. Each set of color points represents a specific 2-car context pattern. In the KITTI dataset, we can observe there are some specific car-to-car "peak" modes in the dataset (similar to the analyses in [25]), while the context patterns are more diverse in the Parking Lot dataset.

4.2 Mining Viewpoint-Occlusion Patterns

As stated above, we present the method of specifying Layer $0-2$ in Fig.2. In this section we present the method of learning viewpoint-occlusion patterns for single cars (i.e., Layer 3 and 4 in Fig.2).

Based on car samples in D^+_{1-car} which do not overlap other cars in images, we specify the single car And-nodes and part Terminal-nodes by learning a mixture of DPMs as done in [8]: (i) cluster the aspect ratios of bounding boxes (used to indicate the latent viewpoints) over D^+_{1-car} to obtain a small number of single car And-nodes and train the initial root appearance templates, and then (ii) pursue the part Terminal-nodes for each single car And-node based on the trained root templates.

Occlusion information is often not available in the car datasets [7,19]. To obtain occlusion information of single cars, we focus on D^+_{2-car} and use overlap ratios between single cars to mine occlusion patterns. In Fig.3 (Middle), we show the two histograms of overlap ratios over D^+_{2-car} plotted on the KITTI [11] and self-collected Parking Lot datasets respectively. In Fig. 3 (Bottom), we show some cropped training positives in the two datasets from which we can observe that overlap ratios roughly reflects the degree of occlusion. Based on the histograms, we mine the viewpoint-occlusion patterns by two methods:

- We adopt the occlusion modeling method proposed in [19] which utilizes car 3D CAD simulation. In addition to the histograms of overlap ratios, we also use the histograms of sizes and aspect ratios of single car bounding boxes to guide the process of synthesizing the occlusion layouts using car 3D CAD models. Then we can learn the And-Or structure for single cars which consists of a small set of consistently visible parts and a number of optional part clusters. Details are referred to [19].

– We cluster the overlap ratios into a small number clusters and each cluster represents an occlusion pattern. The training samples in each cluster are used to train the single car templates and the parts similar to [21,25]. Based on the learned unoccluded single car templates and the estimated threshold using D^+_{1-car}, a car in a car pair is initialized as occluded one if the score is less than the threshold. If the scores of both cars are greater than the threshold, we select the car with lower score as the occluded one. The "unoccluded" car in a car pair is added to D^+_{1-car} if had. Then, we use the same learning method as for D^+_{1-car} except that we only pursue part Terminal-nodes in the "visible" portion of the bounding box of the occluded cars.

5 Learning the Parameters by WLSSVM

In the training data, we only have annotated bounding boxes for single cars. The parse tree pt for each N-car positive sample is hidden. The parameters $\Theta = (\Theta^{app}, \Theta^{def}, \Theta^{bias})$ are learned iteratively. We initialize the parse tree for each N-car positive sample as stated in Sec.4. Then, during learning, we run the DP inference to assign the optimal parse trees for them. We adopt the WLSSVM method [13] in learning. The objective function to be minimized is defined by,

$$\mathcal{E}(\Theta) = \frac{1}{2}\|\Theta\|^2 + C\sum_{i=1}^{M} L'(\Theta, x_i, y_i) \tag{6}$$

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,

$$\begin{aligned} L'(\Theta, x, y) = &\max_{pt\in\Omega_{\mathcal{G}}} [score(x, pt; \Theta) + L_{margin}(y, box(pt))] - \\ &\max_{pt\in\Omega_{\mathcal{G}}} [score(x, pt; \Theta) - L_{output}(y, box(pt))] \end{aligned} \tag{7}$$

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 [13], 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 RHS, so the score of a low-loss prediction gets pulled up. More details are referred to [13,22]. The loss function is defined by,

$$L_{\ell,\tau}(y, box(pt)) = \begin{cases} \ell & \text{if } y = \perp \text{ and pt} \neq \perp \\ 0 & \text{if } y = \perp \text{ and pt} = \perp \\ \ell & \text{if } y \neq \perp \text{ and } \exists\ B \in y \text{ with } ov(B, B') < \tau, \forall B' \in box(pt) \\ 0 & \text{if } y \neq \perp \text{ and } ov(B, B') \geq \tau,\ \forall\ B \in y \text{ and } \exists B' \in box(pt) \end{cases}, \tag{8}$$

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 [14] for our loss formulation.

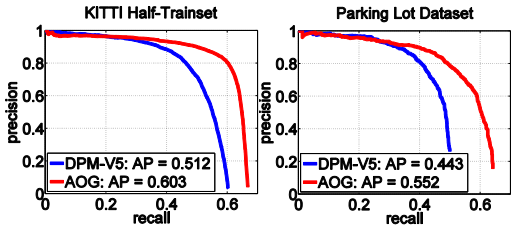

Fig. 4. Precision-recall curves on the test subset splitted from th KITTI trainset (Left) and the Parking Lot dataset (Right)

6 Experiments

6.1 Detection Results on the KITTI Dataset

The KITTI dataset [11] is a recently proposed challenging dataset which provides a large number of cars with different occlusion scenarios. It contains 7481 training images and 7518 testing images, which are captured from an autonomous driving platform. We follow the provided benchmark protocol for evaluation. Since the authors of [11] 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 [8] is a very competitive model with source code publicly available, we compare our model with the latest version of DPM (i.e., voc-release5 [14]). The number of components are set to 16 as the baseline methods trained in [11], other parameters are set as default.

Parameter Settings. We consider N-car with $N = 1, 2$. We set the number of context patterns and viewpoint-occlusion patterns to be 10 and 16 respectively in Sec.4. 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. 2).

Detection Results. The left figure of Fig. 4 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.

Testing on the KITTI Benchmark. We test the trained models above (i.e., using half training set) on the KITTI testset. The detection results and performance comparison are shown in Table 1. This benchmark has three subsets (*Easy, Moderate, Hard*) w.r.t the difficulty of object size, occlusion and truncation. Our model outperforms all the other methods tested on this benchmark. Specifically, our model outperforms OC-DPM [25] on all the three subsets by 5.32%, 1.08%, and 1.74%. We also compare with the baseline DPM trained by ourselves using the voc-release5 code [14], the performance gain of our model

Table 1. Performance comparision (in AP) with baselines on KITTI benchmark [11]

Methods	Easy	Moderate	Hard
mBow [2]	36.02%	23.76%	18.44%
LSVM-MDPM-us [8]	66.53%	55.42%	41.04%
LSVM-MDPM-sv [8,12]	68.02%	56.48%	44.18%
MDPM-un-BB [8]	71.19%	62.16%	48.43%
OC-DPM [25]	74.94%	65.95%	53.86%
DPM (trained by ourselves using [14])	77.24%	56.02%	43.14%
AOG	**80.26**%	**67.03**%	**55.60**%

mainly comes from the *Moderate and Hard* car subsets, with 11.01% and 12.46% in terms of AP respectively. For other DPM based methods trained by the benchmark authors, our model outperforms the best one - MDPM-un-BB by 9.07%, 4.87% and 7.17% respectively.

Note that our model is trained using half of the KITTI trainset, while other methods in the benchmark use more training data (e.g., 1/6 cross validation). The performance improvement by our model is significant. As mentioned by [25], because of the large number of cars in KITTI dataset, even a small amount (1.6%) of AP increasing is still considered significant.

The first 3 rows of Fig. 5 show the qualitative results of our model. The red bounding boxes show the successful detection, the blue ones the missing detection, and the green ones the false alarms. We can see our model is robust to detect cars with severe car-to-car occlusion and clutter. The failure cases are mainly due to too severe occlusion, too small car size, car deformation and/or inaccurate (or multiple) bounding box localization.

6.2 Detection Results on the Parking Lot Dataset

Although the KITTI dataset [11] is very challenging, the camera viewpoints are relatively restricted due to the camera platform (e.g., no birdeye's view), and there is a less number of cars in each image than the ones in parking lot images. Our self-collected parking lot dataset provides more features on these two aspects. As shown in Fig. 5, this dataset has more diversity in terms of viewpoints and occlusions. It contains 65 training images and 63 testing images. Although the number of images is small, the number of cars is noticeably large, with 3346 cars (including left-right mirrored ones) for training and 2015 cars for testing.

Evaluation Protocol. We follow the PASCAL VOC evaluation protocol [7] 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 [11]. We compare with the latest version of DPM implementation [14] and set the number of context patterns and viewpoint-occlusion patterns to be 10 and 18 respectively.

Fig. 5. 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.

Detection Results. In the right of Fig. 4 we compare the performance of our model with DPM. Our model obtains 55.2% in AP, which outperforms the latest version of DPM by 10.9%. The fourth and fifth rows of Fig. 5 show the qualitative results of our model. Our model is capable of detecting cars with different occlusion and viewpoints.

6.3 Detection Results on the Street Parking Dataset

The Street Parking dataset [19] is a recently proposed car dataset with emphases on occlusion modeling of cars in street scenes. We test our model on this dataset to verify the ability of occlusion modeling of our And-Or model. We use two versions of our model for comparison: (i) A hierarchical And-Or model with greedy latent parts, denoting as AOG†, and (ii) A hierarchical And-Or model with visible parts learned based on car 3D CAD simulation, denoting as AOG‡. AOG† and AOG‡ have the same number of context patterns and occlusion patterns, 8 and 16 respectively. To compare with the benchmark methods, we follow the evaluation protocol provided in [19].

Results of our model and other benchmark methods are shown in Table 2, we can see our AOG† outperforms DPM [14] and And-Or Structure [19] by 10.1% and 4.3% respectively. We believe this is because our model takes both context and occlusion into account, and the flexible structure provides more representability of occlusion. Our AOG‡ further improves the performance of AOG†

Table 2. Performance comparision (in AP) on the Street Parking dataset [19]

	DPM [14]	And-Or Structure [19]	our AOG†	our AOG‡
AP	52.0%	57.8%	62.1%	**65.3**%

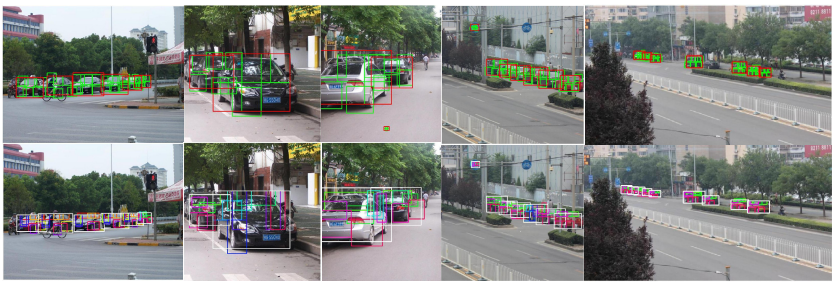

Fig. 6. Visualization of part layouts output by our AOG† (Top) and AOG‡ (Bottom). Best viewed in color and magnification.

by 3.2%, which show the advantage of modeling occlusion using visible parts. The last two rows in Fig. 5 show some qualitative examples. Our AOG is capable of detecting occluded street-parking cars, meanwhile it also has a few inaccurate detection results and misses some cars that are too small or uncommon in the trainset. Fig. 6 shows the inferred part bounding boxes by AOG† and AOG‡. We can observe that the semantic parts in AOG‡ are meaningful, although they may be not accurate enough in some examples.

6.4 Detection Results on the PASCAL VOC2007 Car Dataset

As analyzed by Hoiem, et. al. in [16], cars in PASCAL VOC dataset do not have much occlusion and car-to-car context. We test our And-Or model on the PASCAL VOC2007 car dataset and show that our model is comparable to other single object models. We compare with the latest version of DPM [14]. The APs are 60.6% (our model) and 58.2% (DPM) respectively. We will submit more results in VOC in the future work.

7 Conclusion

In this paper, we propose a reconfigurable hierarchical And-Or model to integrate context and occlusion for car detection in the wild. The model structure is learned by mining context and viewpoint-occlusion patterns at three levels: a) N-car layouts, b) single car and c) car parts. Our model is a DAG where DP algorithm can be used in inference. The model parameters are learned by WLSSVM[13]. Experimental results show that our model is effective in modeling context and occlusion information in complex situations, and obtains better performance over state-of-the-art car detection methods. In our on-going work, we

apply the proposed method to other object categories and study different ways of mining the context and occlusion patterns (e.g., integrating with the And-Or quantization methods [27,18]).

Acknowledgement. B. Li is supported by China 973 Program under Grant no. 2012CB316300 and the National Key Technology Research and Development Program of the Ministry of Science and Technology of China under Grant No. 2014BAK14B03. 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 discussion.

References

1. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 836–849. Springer, Heidelberg (2012)
2. Behley, J., Steinhage, V., Cremers, A.: Laser-based Segment Classification Using a Mixture of Bag-of-Words. In: IROS (2013)
3. Branson, S., Perona, P., Belongie, S.: Strong supervision from weak annotation: Interactive training of deformable part models. In: ICCV (2011)
4. Chen, G., Ding, Y., Xiao, J., Han, T.X.: Detection evolution with multi-order contextual co-occurrence. In: CVPR (2013)
5. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
6. Desai, C., Ramanan, D., Fowlkes, C.: Discriminative models for multi-class object layout. IJCV 95(1), 1–12 (2011)
7. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV (2010)
8. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. TPAMI (2010)
9. Felzenszwalb, P., McAllester, D.: Object detection grammars. Tech. rep., University of Chicago, Computer Science TR-2010-02 (2010)
10. Felzenszwalb, P., Huttenlocher, D.: Distance transforms of sampled functions. Theory of Computing (2012)
11. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: CVPR (2012)
12. Geiger, A., Wojek, C., Urtasun, R.: Joint 3D estimation of objects and scene layout. In: NIPS (2011)
13. Girshick, R., Felzenszwalb, P., McAllester, D.: Object detection with grammar models. In: NIPS (2011)
14. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: Discriminatively trained deformable part models, release 5, `http://people.cs.uchicago.edu/~rbg/latent-release5/`
15. Hejrati, M., Ramanan, D.: Analyzing 3D objects in cluttered images. In: NIPS (2012)
16. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing error in object detectors. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part III. LNCS, vol. 7574, pp. 340–353. Springer, Heidelberg (2012)

17. Hoiem, D., Efros, A., Hebert, M.: Putting objects in perspective. IJCV 80(1), 3–15 (2008)
18. Hu, W., Zhu, S.C.: Learning 3D object templates by quantizing geometry and appearance spaces. TPAMI (to appear, 2014)
19. Li, B., Hu, W., Wu, T.F., Zhu, S.C.: Modeling occlusion by discriminative and-or structures. In: ICCV (2013)
20. Li, B., Song, X., Wu, T.F., Hu, W., Pei, M.: Coupling-and-decoupling: A hierarchical model for occlusion-free object detection. PR 47, 3254–3264 (2014)
21. Mathias, M., Benenson, R., Timofte, R., Van Gool, L.: Handling occlusions with franken-classifiers. In: ICCV (2013)
22. McAllester, D., Keshet, J.: Generalization bounds and consistency for latent structural probit and ramp loss. In: NIPS (2011)
23. Ouyang, W., Wang, X.: Single-pedestrian detection aided by multi-pedestrian detection. In: CVPR (2013)
24. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Teaching 3d geometry to deformable part models. In: CVPR (2012)
25. Pepik, B., Stark, M., Gehler, P., Schiele, B.: Occlusion patterns for object class detection. In: CVPR (2013)
26. Sadeghi, M., Farhadi, A.: Recognition using visual phrases. In: CVPR (2011)
27. Song, X., Wu, T.F., Jia, Y., Zhu, S.C.: Discriminatively trained and-or tree models for object detection. In: CVPR (2013)
28. Tang, S., Andriluka, M., Schiele, B.: Detection and tracking of occluded people. In: BMVC (2012)
29. Tu, Z., Bai, X.: Auto-context and its application to high-level vision tasks and 3D brain image segmentation. TPAMI (2010)
30. Yang, Y., Baker, S., Kannan, A., Ramanan, D.: Recognizing proxemics in personal photos. In: CVPR (2012)
31. Zhu, L., Chen, Y., Yuille, A., Freeman, W.: Latent hierarchical structural learning for object detection. In: CVPR (2010)
32. Zhu, S.C., Mumford, D.: A stochastic grammar of images. Found. Trends. Comput. Graph. Vis. (2006)