Recognizing Car Fluents from Video

Bo Li[1,*], Tianfu Wu[2], Caiming Xiong[3,*] 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   [3]Metamind Inc.

boli86@bit.edu.cn, {tfwu, sczhu}@stat.ucla.edu, cmxiong@metamind.io

Abstract

Physical fluents, a term originally used by Newton [40], refers to time-varying object states in dynamic scenes. In this paper, we are interested in inferring the fluents of vehicles from video. For example, a door (hood, trunk) is open or closed through various actions, light is blinking to turn. Recognizing these fluents has broad applications, yet have received scant attention in the computer vision literature. Car fluent recognition entails a unified framework for car detection, car part localization and part status recognition, which is made difficult by large structural and appearance variations, low resolutions and occlusions. This paper learns a spatial-temporal And-Or hierarchical model to represent car fluents. The learning of this model is formulated under the latent structural SVM framework. Since there are no publicly related dataset, we collect and annotate a car fluent dataset consisting of car videos with diverse fluents. In experiments, the proposed method outperforms several highly related baseline methods in terms of car fluent recognition and car part localization.

1. Introduction

1.1. Motivation and Objective

The term of physical fluents is first introduced by Newton [40] to represent the time-varying statuses of object states. In the commonsense literature, it is defined as the specifically varied object status in a time sequence [39]. As a typical instance, car fluent recognition has applications in video surveillance and self-driving. In autonomous driving, car fluents are very important to infer the road conditions and the intents of other drivers. The left of Fig. 1 shows an example, when seeing the left-front door of a roadside car is opening, the autonomous car should slow down, pass with cautions, or be ready to stop. In the freeway context, the autonomous car should give way when the front car is blinking to require a left or right lane change (see the middle figure in Fig. 1). In the parking lot scenario, car part

Figure 1. Some important applications of car fluent recognition. Best viewed in color and zoom in.

status change indicates info for reasoning about human-car interactions and surveillance, e.g., a woman is opening car trunk to pick up stuff (see the right of Fig. 1). In general, fluents recognition is also essential in inferring the minds of human and robotics in Cognition, AI and Robotics.

While there is a vast literature in computer vision on vehicle detection and viewpoint estimation [16, 21, 45, 46, 63, 43, 25, 35, 12], reasoning about the time-varying states of objects (i.e., fluents) has been rarely studied. Car fluents recognition is a hard and unexplored problem. As illustrated in Fig. 2, car fluents have large structural and appearance variations, low resolutions, and severe occlusions, which present difficulties at multiple levels. In this paper, we address the following tasks in a hierarchical spatial-temporal And-Or Graph (ST-AOG).

i) Detecting cars with different part statuses and occlusions (e.g., in Fig.2 (1.b), the frontal-view jeep with hood being open and persons wandering in the front. In the popular car datasets (such as the PASCAL VOC car dataset [15] and KITTI car dataset [20]), most cars do not have open parts (e.g., open trunk/hood/door) together with car-person occlusions. So, a more expressive model is needed in detecting cars in car fluent recognition.

ii) Localizing car parts which have low detectability as individual parts (e.g., the open hood and trunk in Fig. 2 (2.c), the tail lights in the last row of Fig. 2) or even are occluded (e.g., the trunk in Fig. 2 (2.b)). This situation is

---
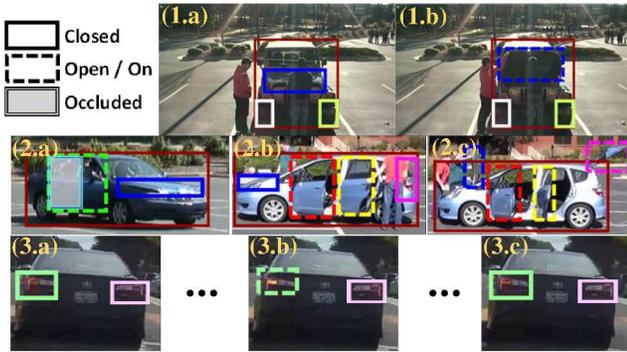
Figure 2. Illustration of the large structural and appearance variations and heavy occlusions in car fluent recognition. We focus on the fluents of functional car parts (e.g., doors, trunks, lights) in this paper. For clarity, we show the cropped single cars only. See Fig. 1 and Fig. 5 for the whole context. See text for details.

similar, in spirit, to the well-known challenge of localizing hands/feet in human pose recovery [67]. Temporal contextual information is needed to localize those parts, besides a spatial hierarchical car pose model.

iii) Recognizing time-varying car part statuses is a new task. Unlike object attributes (such as gender and race) which are stable for a long time, the time-varying nature of car fluents presents more ambiguities (e.g., in Fig. 2 (3.a-3.c), the periodical status change of car lights, and its ambiguous appearance and geometric features). Unlike action recognition which focuses on humans and does not account for the pose and statuses of parts [31, 58], car fluent recognition is fine-grained in both spatial and temporal domains.

In this paper, we propose to learn a ST-AOG [71, 21, 10, 24] to represent car fluents at semantic part-level. Fig. 3 shows a portion of our model, the ST-AOG span in both spatial and temporal dimensions. In space, it represents the whole car, semantic parts, part status from top to bottom. In time, it represents the location and status transitions of the whole car and car parts from left to right. Given a test video, ST-AOG will output frame-level car bounding boxes, semantic part (e.g, door, light) bounding boxes, part statuses (e.g., open/close, turn on/off), and video-level car fluents (e.g., opening trunk, turning left).

Because of the lateral edges in Fig. 3, our proposed ST-AOG is no longer a directed acyclic graph (DAG), thus dynamic programming (DP) cannot be directly utilized. To cope with this problem, we incorporate loopy belief propagation (LBP) [62] and DP in our inference, and adopt a part-based hidden Markov Model (HMM) in temporal transition for each semantic part (see Fig. 4). All the appearance, deformation and motion parameters in our model are trained jointly under the latent structural SVM framework.

To train and evaluate the ST-AOG, we collect and annotate a car fluent dataset due to the lack of publicly available benchmark (Fig. 5 shows some examples, and details of the dataset will be given in Section 6.1). Experimental results show that our model is capable of jointly recognizing semantic car fluents, part positions, and part status in video. It outperforms several highly related state-of-the-art baselines in fluent recognition and part localization (sliding window-based). Besides, our model can be directly incorporated with popular improved dense trajectories (iDT) [59] and C3D features [55] to encodes more general information.

2. Related Work and Our Contributions

We briefly review the most related topics in 3 streams:

Car Detection and View Estimation. In computer vision and intelligent transportation, there is a considerable body of work on car detection and view estimation [16, 21, 45, 46, 63, 43, 54, 25, 35, 12]. Though those works have successfully improve the performance on popular benchmarks [15, 20], the output is usually the car bounding box and a quantized rough view. Some other works [53, 72, 65] aim to get car configurations with detailed or fine-grained output to describe the more meaningful car shape, rather than a bounding box. However, all of those works generally regarded car as a static rigid object, while pay little attention to the functionalities of semantic car parts and the fact that cars presenting large geometry and appearance transformations during car fluents change.

Video-based Tasks in Vision, Cognition, AI and Robotics. In computer vision, a significant effort has been devoted to video-based tasks: event [38, 28, 64], action detection/recognition [61, 73, 60, 41] and pose estimation [67, 1, 18]. These papers are related to our work, however, most of them are based on human models, while very little work has done on car fluents and related car part status estimation. In cognitive science, the concept of fluents is mainly used to represent the object status changing in time series [39], and it has been used in causality inference [39, 26, 7, 19]. Furthermore, fluents are also related to action-planning in AI and Robotics [5, 66].

Human-Object Recognition. As mentioned in [19], the status of object fluent change is caused by some other objects. In our case, the car fluent change is mainly conducted by humans. Thus our work is related to part recognition [49, 34] and human-object interactions [50, 33, 23, 68, 6, 51, 14]. But, here we jointly and explicitly model car detection, part localization and status estimation.

Our Contributions. This paper makes three contributions to car fluent recognition:

i) It presents a ST-AOG to represent the spatial-temporal context in car fluent recognition. Car fluent recognition is a new task in the computer vision literature.

ii) It presents a new car fluent dataset with per-frame annotations of cars, car types, semantic parts, part statuses and viewpoints.

iii) It provides a new method for car fluent recognition and part status estimation, and outperforms several state-of-the-art baseline methods.
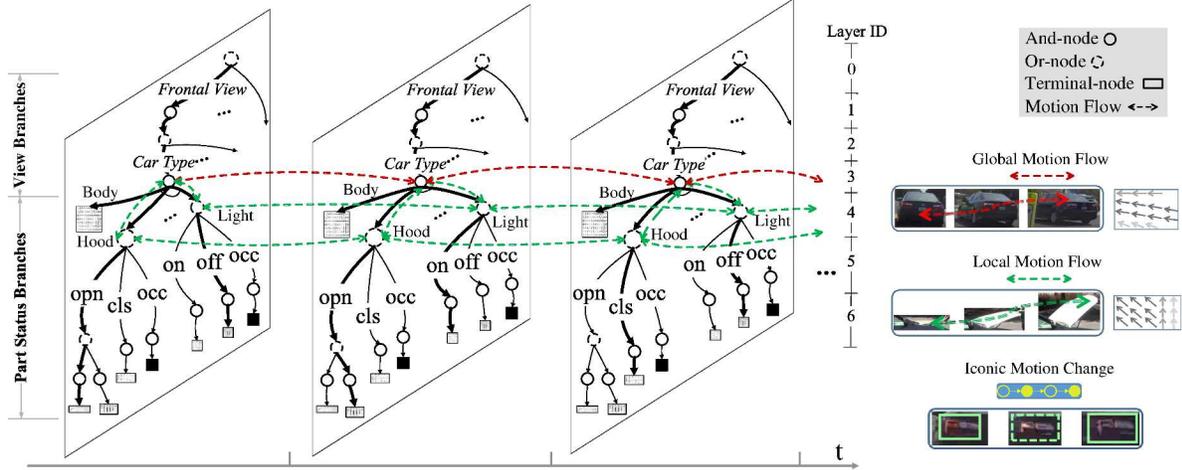
Figure 3. Illustration of the proposed ST-AOG for representing car fluents. In space (vertical), the ST-AOG represents cars by quantized views, types, and semantic parts with different statuses. This is used to model appearance and geometry info. In time (horizontal), the ST-AOG represents temporal transitions by 3 type of motion flows: Global motion flow, local motion flow and iconic motion change, which are illustrated on the right. Here, "opn", "cls", "occ", "on", "off" denotes "open", "close", "occluded", "turn on", "turn off", respectively. For each fluent video, our ST-AOG outputs a parse graph consists of frame-level parse trees, which are shown by the bold arrows. We omit the temporal transitions of Terminal-nodes for clarity.

3. Representation

3.1. Spatial-Temporal And-Or Graph

Fig. 3 illustrates our ST-AOG, it models a car fluent by the viewpoint, car type, semantic parts and corresponding part statuses hierarchically. In addition to the appearance and deformation, it also models the temporal transitions of the semantic parts.

Utilizing I_i and I_{i+1} to denote neighbouring frames, the ST-AOG is defined by a 3-tuple,

$$\mathcal{G} = (V, E, \Theta) \qquad (1)$$

where V is the node set consisting of a set of non-terminal nodes (i.e., And-nodes V_{And} and Or-nodes V_{Or}) and a set V_T of terminal-nodes (i.e., appearance templates). $E = (E_{\mathcal{S}}, E_{\mathcal{M}})$ is the edge set maintaining the structure of the graph. Specifically, $E_{\mathcal{S}}$ is the spacial edges accounting for composition and deformation, $E_{\mathcal{M}}$ is the lateral edges accounting for temporal transitions. $\Theta = (\Theta^{app}, \Theta^{def}, \Theta_{\mathcal{M}}, \Theta^{bias})$ is the set of parameters, and we have,

i) the appearance parameters $\Theta^{app} = (\Theta_t^{app}, t \in V_T)$ consists of the appearance templates for all terminal-nodes. Our model is general, in which any type of features can be used. In this paper, we use the histogram of oriented gradient (HOG) feature [13] and convolutional neural network (CNN) feature [32] as the appearance feature for illustration.

ii) the deformation parameters Θ^{def} are defined for all in-edges of all terminal-nodes, $(v, t) \in E$ ($v \in V_{And} \cup V_{Or}$ and $t \in V_T$), to capture the local deformation when placing a terminal-node t in a feature pyramid. For the deformation feature, we adopt the standard separable quadratic function of the displacement of t with respect to its parent

node v, (dx, dx^2, dy, dy^2), as done in the deformable part-based models [16, 2] and the car AOG model [37].

iii) the transition parameters $\Theta_{\mathcal{M}}$ are defined on the single car and semantic part nodes. They are used to weight the effects of temporal transition between I_i and I_{i+1}. As illustrated on the right of Fig. 3, we compute 3 types of motion flows for car fluents:

a) global motion flow. We compute the optical flow of the whole car (e.g., body), and utilize it as the global feature to localize car in time. This global feature is very helpful for moving cars, especially in the far view, in which the semantic part is too small to be detected.

b) local motion flow. We compute the optical flow of "mobile" car part (e.g., door), and utilize it as the temporal feature to recognize the position and status of the part. Here, the "mobile" car part is the part that can be open or closed, which has motion features during fluent change.

c) iconic motion flow. For car lights, the most distinctive feature is its intensity change and temporally periodical turn on/off statuses. This is related to the work in iconic change [3, 4] and change detection [47]. Here, we use the appearance and optical flow templates to localize car lights, and compute the temporal intensities (vector of normalized intensities) to recognize the periodical light status change (e.g., left-turn/right turn).

For optical flow, we compute the pyramid feature to cope with different scales.

iv) the bias parameters Θ^{bias} are defined for the child nodes of all Or-nodes to encode either the prior (e.g., different appearance templates for the "open" part), or the compatibility of parsing scores among different sub-categories (e.g., different viewpoints of a jeep).

Although the hierarchical structure of our ST-AOG is similar to those spatial-only models used in [37, 46] for car detection, we introduce the semantic part Or-nodes (to be used to define fluent) for detailed part status modelling in both spatial and temporal domains.

A **parse tree**, pt, is an instantiation of the ST-AOG being placed at a location in the spatial-temporal feature pyramid. It is computed by following the breadth-first-search order of the nodes in $\mathcal{G}$, selecting the best child node at each encountered Or-node (based on the scoring function to be defined later on). The bold black edges in Fig. 3 show the parse trees of ST-AOG on three neighbouring video frames. For a video with N frames, we can get its **parse graph** as $pg = \{pt^{i,i+1}\}_{i=1}^{N-1}$.

Based on the pg, we extract frame-level part bounding boxes and statues, and utilize them to design spatial-temporal features for fluent recognition. To capture long-term spatial-temporal info, our model can also integrate iDT [59] and C3D features [55] (see Section 6.2).

3.2. The Scoring Function

The scoring function of ST-AOG is recursively defined w.r.t. its structure. For simplicity, we will use v and $\widetilde{v}$ ($v \in V$) to represent the temporally neighbouring nodes (on frames I_i and I_{i+1}) below.

Let $O \in V_{Or}$ denotes the Or-node in the ST-AOG, $A \in V_{And}$ be the parent node of a terminal-node $t \in V_T$. We model t by a 4-tuple $(\theta_t^{app}, \theta_{t|A}^{def}, \sigma_t, a_{t|A})$ where σ_t is the scale factor for placing t w.r.t. A in the feature pyramid ($\sigma_t \in \{0, 1\}$), and $a_{t|A}$ is the anchor position of t relative to A.

i) Given $A, \widetilde{t}$ and their locations $l_A, l_{\widetilde{t}}$, *the scoring function* of placing t at the position l_t is then defined by,

$$S(t|l_{\widetilde{t}}, A, l_A) = \max_{l_t}[< \theta_t^{app}, \Phi^{app}(l_t) > -$$
$$< \theta_{t|A}^{def}, \Phi^{def}(l_t, l_A) > + \theta_{\mathcal{M}}^T \|l_t - l_{\widetilde{t}} + \mathcal{F}(l_t)\|_2^2] \quad (2)$$

where $\Phi^{app}(l_t)$ is the appearance features (HOG or CNN) and $\Phi^{def}(l_t, l_A)$ is the deformation features. $\theta_{\mathcal{M}}^T$ is the motion flow weight of t. $\mathcal{F}(l)$ is the motion flow between frames I_i and I_{i+1} computed at position l.

ii) Given $\widetilde{A}$ and its location $l_{\widetilde{A}}$ at frame I_{i+1}, *the scoring function of A* is defined by,

$$S(A, l_A|\widetilde{A}, l_{\widetilde{A}}) = \sum_{v \in ch(A)} [S(v|A, l_A) + b_A$$
$$+ \theta_{\mathcal{M}}^A \|l_{v|A} - l_{v|\widetilde{A}} + \mathcal{F}(l_{v|A})\|_2^2] \quad (3)$$

where $ch(A)$ is the set of child nodes of A, b_A is the bias parameter. $\theta_{\mathcal{M}}^A$ is the motion flow weight of A.

iii) Given $\widetilde{O}$ and its location $l_{\widetilde{O}}$ at frame I_{i+1}, *the scoring function of O* is defined by,

$$S(O, l_O|\widetilde{O}, l_{\widetilde{O}}) = \max_{v \in ch(O)} [S(v|O, l_O) +$$
$$\theta_{\mathcal{M}}^O \|l_{v|O} - l_{v|\widetilde{O}} + \mathcal{F}(l_{v|O})\|_2^2] \quad (4)$$

where $ch(O)$ is the set of child nodes of O, l_O is the position of O. $\theta_{\mathcal{M}}^O$ is the optical flow weight of O.

For **temporal term** $\|l_v - l_{\widetilde{v}} + \mathcal{F}(l_v)\|_2^2, v \in V$, since the scoring function of v is conditioned on $\widetilde{v}$ in time, while $\widetilde{v}$ is conditioned on its child nodes or parent node in space, there are loops in this structure, and we couldn't use DP to compute above scoring functions directly. As similar to [11], we resort to LBP [62] to get an approximate solution. For computational efficiency, the motion flows are computed only at nodes corresponds to the car and its semantic parts. For other nodes, their motion flow are implicitly embedded, as their score maps are related to nodes with temporal links, and their scoring functions are computed as above but without the temporal terms.

Given a fluent video clip with N frames, the overall score of ST-AOG will be:

$$S(I_{1:N}|pg, \mathcal{G}) = \frac{1}{N-1} \sum_{i=1}^{N-1} S^{i,i+1}(O, l_O|\widetilde{O}, l_{\widetilde{O}}) \quad (5)$$

where O is the root Or-node in ST-AOG. In probabilistic model, Eqn. (5) can be interpreted as a log-posterior probability up to a constant. Our objective is to maximize Eqn. (5), given all appearance, deformation, and motion features from training videos.

4. Inference by LBP and DP

To cope with the loop introduced by motion transition, we extend the traditional inference procedure in AOG [37, 69] with LBP. Given an input video, Our inference procedure includes 4 steps:

i) For each frame I_i, we omit the temporal links between it and its neighbour frames, and compute the appearance feature pyramid, optical flow pyramid, and score maps for all nodes in Layer $3 - 6$ in the ST-AOG by the Depth-First-Search (DFS) order; This step is similar to the inference algorithm in [37].

ii) After we get the score maps of the semantic part Or-nodes and the single car And-nodes, we further integrate the score maps with optical flow maps, as can be seen on the left of Fig. 4. For each semantic part p with its parent root node r, we focus on four nodes, i.e., $p, r, \widetilde{p}, \widetilde{r}$, and omit other spatial links connected to them. At the beginning, we send message from p to r, then from r to the rest to update the message. When the difference of the propagated message in two consecutive iterations doesn't change, we compute the last "belief" that transmitted to r as $r's$ score map. This procedure can be efficiently implemented by distance transform [17].

iii) After we get the score maps from *ii)*, we further compute the score maps for nodes in the upper layers of ST-AOG. By this procedure, the score map of the root Or-node for neighbouring frames I_i and I_{i+1} can be computed by maximizing Eqn. (4) on each spatial-temporal point.
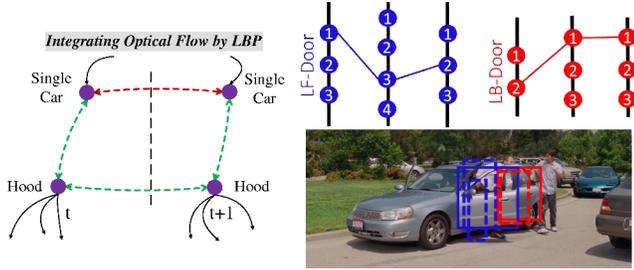
Figure 4. Left: Computing optical flow with LBP. For simplicity, we just show the flow circle of "Body-Hood". Right: Semantic part inference with part-based HMM. Here, each bold line represents a frame, circles represent part proposals with different status, thin lines represent the temporal transitions, and semantic parts are represented by different colors. For simplicity, we just show the bounding box proposals of the left-front door and left-back door. Best viewed in color and zoom in.

iv) For each spatial-temporal sliding window with score being greater than the threshold τ, we follow the Breadth-First-Search (BFS) order to retrieve its parse tree (including the whole car windows, semantic part windows and part statuses at each frame) by taking `argmax` at each encountered Or-node.

4.1. Post-Refinement

As analysed in [44], we can get more accurate part localizations by generating multiple object instances in an image. In this paper, we utilize a similar method to generate multiple car part proposals: First of all, we set low detection threshold to generate multiple car proposals around the target car, then we execute a conservative non-maximum suppression (NMS) procedure to override the observation term and select windows that are not local maxima. By backtracing the parse tree pt, we can get the semantic part proposals from each car proposal.

The right of Fig. 4 illustrates our part-based HMM inference method. For a semantic part p, we can write its bounding box sequence in a given video (N frames) as a stochastic set $\Omega_p = \{p^i\}_{i=1}^{N-1}$, then the overall linking scores of part p in a video can be computed as:

$$S(\Omega_p) = \sum_{i=1}^{N-1} S(p^i) + \theta_{\mathcal{M}}^p \psi(p^{i,i+1}) + \lambda ov(p^i, p^{i+1}) \quad (6)$$

$$S(p^i) = \theta_p \phi_p(I_i) + \theta_p \phi_p(I_{i+1}), \psi(p^{i,i+1}) = (dx, dx^2, dy, dy^2, ds, ds^2)$$

where we model $S(p^i)$ as the appearance and deformation score, θ_p and ϕ_p are the corresponding parameter and feature, $\theta_{\mathcal{M}}^p$ is the temporal weights, ψ is the temporal feature, dx, dy, ds are the differences of temporal position and scale of part p between I_i and I_{i+1}, $ov(p^i, p^{i+1})$ is the bounding box overlap of p between I_i and I_{i+1}. For each semantic part in the video, we seek the optimal path Ω_p^* by maximizing Eqn. (6).

5. Learning by Latent Structural SVM

The training data. In our car fluent dataset, for each video $X_v, v \in \{1, \cdots, W\}$, we have its annotated parse graph $pg(X_v)$ including the car bounding boxes, viewpoints, car types, semantic part bounding boxes, part statuses and corresponding fluent labels.

For parameter optimization, the objective function of ST-AOG is:

$$\min_{\Theta, \xi} \quad \mathcal{E}(\Theta) = \frac{1}{2}\|\Theta\|^2 + C \sum_{v=1}^{W} \xi_v \quad (7)$$

$$\text{s.t.} \quad \xi_v > 0; \quad \forall v, \quad \widehat{pg}(X_v) \neq pg(X_v),$$

$$S(pg(X_v)) - S(\widehat{pg}(X_v)) \geq L(pg(X_v), \widehat{pg}(X_v)) - \xi_v$$

where C is the regularization parameter, $\widehat{pg}$ is the predicted parse graph, and $L(pg(X_v), \widehat{pg}(X_v))$ is the surrogate loss function. To accurately predict the viewpoint, car type, part bounding boxes and part statuses, we specify $L(pg(X_v), \widehat{pg}(X_v)) = \frac{1}{v_N} \sum_{j=1}^{v_N} \ell(pt(X_v^j), \widehat{pt}(X_v^j))$ (v_N is the number of frames in X_v), and define $\ell(pt, \widehat{pt})$ by,

$$\ell(pt, \widehat{pt}) = \begin{cases} 1 & \text{if view or car type is different} \\ 1 & \text{if } overlap(pt, \widehat{pt}) < ov \\ 1 & \text{if } \exists p, pt.status(p) \neq \widehat{pt}.satus(p) \\ 0 & \text{otherwise} \end{cases} , \quad (8)$$

where the second term computes the overlap between two parse trees (they have the similar structure, otherwise the loss is computed based on the first term), and $overlap(pt, \widehat{pt})$ is defined by the minimum overlap of bounding boxes of all nodes in the parse trees. The third term check the semantic part status.

The training procedure has the following two steps:

i) Initialization. We use standard SVM to train the semantic car part templates on the annotations. For each view, we first cluster the semantic parts by their status, then cluster the "open" parts by their aspect ratios, as they have large appearance and geometry variations. After getting these clusters, we train a template for each of them. These templates construct the Terminal-nodes in ST-AOG.

ii) Learning the whole ST-AOG under the LSSVM. As the groundtruth annotations may not be the most discriminative, we allow parts to move around but with a certain overlap with the groundtruth. Thus the optimization becomes a latent structural problem. We iterate the following two steps to get a local optima:

Step 1 - Positives Relabelling: Do loss-augmented inference for each training video with current model by:

$$\Omega_p^* = \arg\max \sum_{p=1}^{P} S(\Omega_p) + L(pg(X_v), \widehat{pg}(X_v)) \quad (9)$$

Step 2 - Update Weights: With the inferred part bounding box and part status sequences, we use standard Structural SVM to optimize the appearance, deformation, and motion parameters Θ in ST-AOG.

In our implementation, we start with a small number of negative examples, and iteratively increase them in the 2nd step as the iCCCP method [70]. Besides, we also maintain a cache for hard negative mining. Since computing on every frame is not necessary, we sliding the spatial-temporal window by every 3 frames.

Figure 5. Sample images in Car-Fluent dataset.
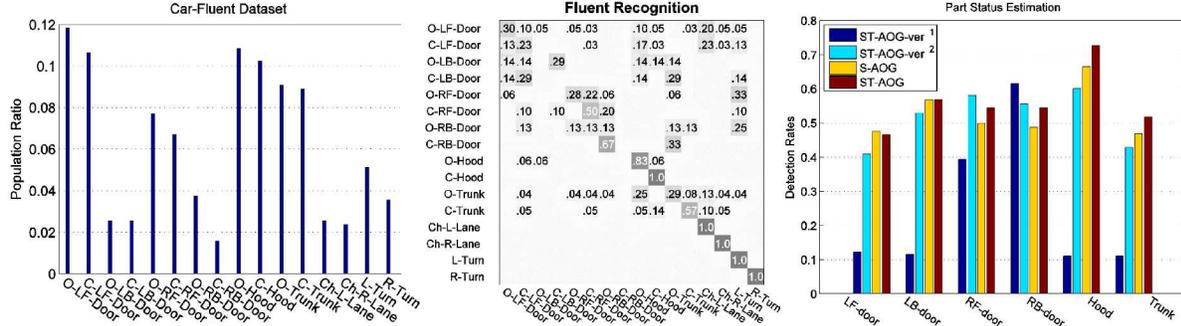
Figure 6. Left: semantic fluents and their population ratios on Car-Fluent Dataset. Middle: confusion matrix of the best ST-AOG on this dataset. Right: part status estimation results of different variants of ST-AOG. Here, "O", "C", "Ch", "L", "R" denotes "open", "close", "change", "left", "right", respectively.

6. Experiments

6.1. Car-Fluent Dataset

The introduced Car-Fluent dataset includes 647 video clips, containing basically 16 semantic car part fluents with multiple car types, camera viewpoints and occlusion conditions. All the fluents and its population ratios are shown on the left of Fig. 6. The videos in Car-Fluent dataset are collected from various sources (youtube, movies, VIRAT [42], etc.), and captured by both static cameras and moving cameras. Based on the number of semantic fluent instances in each video, we split these videos into 2 sets: the first set consists of 507 videos where each video contains a single instance of each car fluent. We randomly select 280 videos in this dataset for training, and the rest for testing; The second split consists of 140 videos that are just used for testing, in which each video is much longer and has multiple instances of diverse kinds of car fluents. All the videos are used for three important tasks: 1) semantic fluent recognition (e.g. opening hood, change left lane), 2) car part localization (e.g. where is the hood? Where is the light?), and 3) car part status estimation (e.g., is the trunk open?).

All videos are annotated with viewpoints, car types, bounding boxes of both whole car and semantic parts, and part statuses. The type of part status are *"open/on"*, *"close/off", and "occluded"*. In this paper, we focus on "semantic" and "functional" parts, all these parts are related to the fluents listed in Fig. 6. We adopt VATIC [57] to annotate videos on the introduced dataset. 7 workers are hired to clip the sequence of semantic car fluents and annotate the videos. Three independent annotations were collected for each video and combined via outlier rejection and averaging. It took them about 500 hours in total. Finally, all labelling results were thoroughly checked by the authors. For more details, please refer to our supplementary.

6.2. Fluent Recognition

To verify our ST-AOG on fluent recognition, we compare it with space-time interest points (STIP) [31], improved dense trajectory (iDT) [59], and the deep learning based C3D method [55]. For STIP and iDT, we use the softwares in [31, 59] to extract corresponding features inside the car bounding boxes from each video, and try various methods to construct the video descriptors. For the bag-of-words (BoW) method, we try different size of codebooks (i.e., $400, 800, 1000, 1200, 1600, 2000$), and select the best ones for iDT and STIP. For fisher vectors (FV) and VLAD vectors, we use 256 components for as common choices in [64, 52, 29]. For feature dimension reduction, we use PCA, and experiment with different reduced dimensions, i.e., $1/4, 1/3, 1/2, 1$ of the original dimension, and utilize the reduced dimensions that best balance the performance and storage cost in corresponding features, i.e., half of the original dimension for both STIP and iDT.

For C3D [55], since we don't have enough videos to train a deep network, we first pre-train a network on the Sport1M dataset [30] and VIRAT dataset [42]. Then we finetuned a model on the Car-Fluent dataset. As in [55], we extract fc6 features from each video and use the mean features as the video descriptors. We also tried computing the FV and VLAD vectors for C3D method, but get low performance.

For our ST-AOG, based on the predicted semantic part positions and part statuses extracted from parse graph pg, we compute temporal part status (TPS) features as follows:

$$\varphi_T = \{[\phi_1(p_s^i), \phi_2(p_s^i, p_s^{i+1}), d(p^i, p^{i+1})]_{p=1}^P\}_{i=1}^{N-1}$$

where $\phi_1(s) = [0, \cdots, 1(s), \cdots, 0]$ is a z_p dimension vector, here z_p is the number of statuses of part p, $\phi_2(s^i, s^{i+1}) = [0, \cdots, 1(s^i)1(s^{i+1}), \cdots, 0]$ is a $z_p \times z_p$ dimension vector, $d(p^i, p^{i+1})$ is the temporal transition of
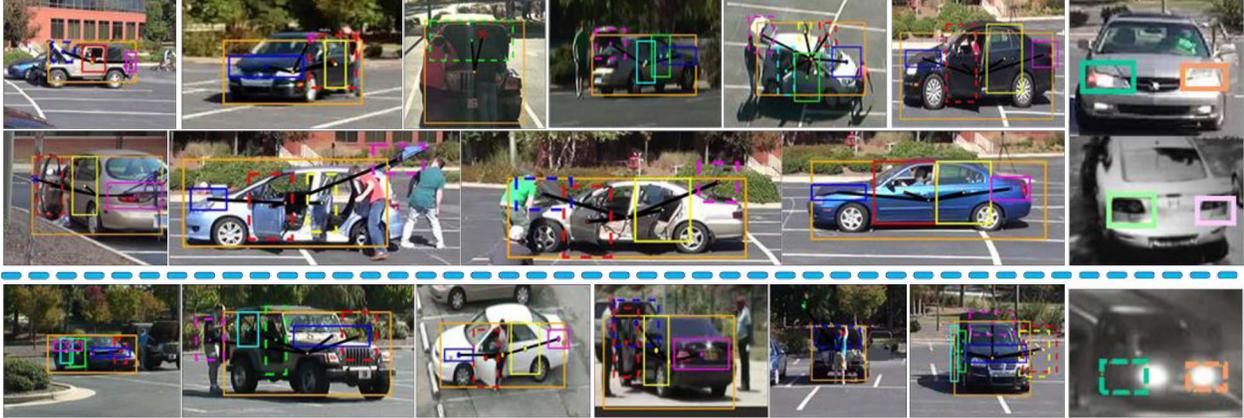
Figure 7. Examples of part status estimation results on our Car-Fluent dataset. For good visualization, we cropped the cars from original images. Different semantic parts are shown by rectangles in different colors (solid for "close/off" status and dashed for "open/on" status). The black lines are used to illustrate variations of the relative positions between whole car and car parts with different statuses. The top 2 rows show successful examples. The bottom row shows failure cases, which are mainly caused by occlusion and mis-detected car view. by failure, we mean some part statuses are wrong (e.g., for the 3rd example, the door is correct, but the hood and trunk are wrong). Best viewed in color and magnification.

	BaseLines							ST-AOG		
	STIP-BoW	iDT-BoW	STIP-FV	STIP-VLAD	iDT-FV	iDT-VLAD	C3D	TPS	TPS+iDT	TPS+iDT+C3D
MP	28.8	35.1	35.2	31.5	40.7	41.3	31.0	34.4	45.0	**50.8**

Table 1. Results of baseline models and our ST-AOG in terms of mean precision (MP) in **Car Fluent Recognition**.

part p between neighbouring frames. For each car light, we also compute the mean intensity difference over the light bounding box across neighbouring frames, and append it to φ_T. We compute FV and VLAD vectors on TPS to construct the video descriptors as STIP and iDT. For our TPS, we get the best performance with 53 reduced feature dimensions for PCA and 256 centers for VLAD vectors.

We use the one-against-rest approach to train multi-class SVM classifiers for above methods. For evaluation, we plot confusion matrix, and report the mean precision (MP) in fluent recognition, equivalent to the average over the diagonal of the confusion matrix.

The first 6 columns in Table 1 show the MP of above methods. As can be seen, the "iDT-VLAD" get the best performance, while C3D and our TPS get relatively lower performances on this dataset, which verified the importance of motion features. For C3D, we think the relatively low performance is due to the lack of training data and it doesn't take advantage of car positions. As TPS, iDT and C3D belong to different type of features, we augment our TPS by integrating it with iDT and C3D features. To this end, we first combine our TPS with iDT, specifically, we try various combinations of FV and VLAD vectors, and find the best one is concatenating TPS VLAD vectors and iDT VLAD vectors with square root normalization, as can be seen in Table 1, this improves the performance by about 4; then we further combine TPS and iDT VLAD vectors with C3D feature, as can be seen in the last column, we improve the performance to 50.8, which outperforms the best single feature by 9.5. From these results we can see, ST-AOG provides

important cues in car fluent recognition, and it's very general to integrate with traditional motion features and deep learning features.

The middle of Fig. 6 shows the confusion matrix of our "TPS+iDT+C3D" method. We can see the merged feature performs pretty well on fluents, i.e., "Open/Close Hood", "Change Left/Right Lane", "Left/Right Turn", but totally miss fluents, e.g., "Open/Close LB Door", "Close RB Door". For fluents related to the car back door, the low performance may be attributed to their ambiguous features and relatively low population ratios on current dataset.

6.3. Part Localization and Part Status Estimation

Since part location and status are important to fluent recognition, in this experiment, we further examine the frame-level part localization and status estimation results.

6.3.1 Part Localization

For semantic part localization, we use following baselines: **(i)** *Strongly supervised deformable part-based model (SS-DPM) [2]*. SSDPM is an open source software, it achieved state-of-the-art performance on localizing semantic parts of animals on PASCAL VOC 2007 dataset [15]. **(ii)** *And-Or Structures [36]*. In the literature, [36] also attempted to localize the semantic parts by replacing the latent ones in DPM. **(iii)** *Deep DPM (DP-DPM) [22]*. The authors in [22] argued that by replacing HOG features with Deep CNN features can improve the detection performance of original DPM [16]. We use the same trick as [36] to add semantic parts on DP-DPM. We also try to compare our model with [67, 27, 8, 9], but find it's hard to compare with them,

	SSDPM [2]	AOG-Car [36]	DP-DPM [22]	ST-AOG-HOG	ST-AOG-HOG-CSC	ST-AOG-CNN
Body	96.4	**99.1**	93.4	99.0	90.8	94.7
Left-front door	36.3	36.6	14.4	**49.9**	48.0	36.6
Left-back door	42.2	44.4	7.1	**60.3**	55.1	32.3
Right-front door	31.6	40.7	13.0	**58.4**	55.9	26.8
Right-back door	33.0	19.4	14.2	**61.4**	55.3	32.8
Hood	35.9	20.1	7.1	**73.7**	67.5	38.4
Trunk	25.4	16.8	10.1	**54.0**	49.7	33.7
Left-head Light	10.4	22.1	19.6	**33.1**	27.8	29.3
Right-head Light	13.3	24.3	15.7	**41.3**	36.5	23.3
Left-tail Light	6.8	25.7	18.3	**27.8**	27.3	22.7
Right-tail Light	6.2	**31.1**	14.0	23.2	22.9	14.0
Mean	24.1	28.1	13.4	**48.3**	44.6	29.0

Table 2. Semantic **Car Part Localization** results of baseline methods and ST-AOG on Car-Fluent dataset.

as either the code is not available, or the design of parts is inappropriate in our case (details are in the supplementary).

For our ST-AOG, we use several variants for comparison. 1)"ST-AOG-HOG" is the instance of our ST-AOG based on HOG features. 2) "ST-AOG-HOG-CSC" is the cascade version of our model. To improve the speed, we use the recently developed Faster-RCNN [48] for cascade detection of cars. Then based on these car detections, we further run the ST-AOG for part localization. 3) "ST-AOG-CNN" is the instance of our ST-AOG based on the pyramid of CNN features, here, we use the max_5 layer of the CNN architecture for comparison as [22]. To cope with small cars, we double the size of source images for CNN.

For evaluation, we compute the detection rate of the semantic parts (e.g., "left-front door", "hood"). Here, we assume the whole car bounding boxes are given, and each method only outputs 1 prediction with the highest confidence for each groundtruth car. Table 2 shows the quantitative results. We can see on almost all semantic parts, our model outperforms baseline models by a large margin. We believe this is because ST-AOG jointly model the view, part occlusion, geometry and appearance variation, and temporal shifts. For "ST-AOG-HOG-CSC", we just use it as a reference to improve the speed of ST-AOG, since it has the nearly real-time speed, and can be plugged in our model as an outlier-rejection module. Surprisingly, CNN features perform worse than HOG features on the Car-Fluents dataset. In experiments, we find the extracted CNN feature on deep pyramid [22] is too coarse (the cell size is 16), even we resize original images by 2 times, it still miss many small parts. Based on recent study of using CNN for keypoints prediction [56], we believe a more effective method is required to integrate CNN feature with graphical models.

6.3.2 Part Status Estimation

To the best of our knowledge, no model tried to output car part status in our case in the literature. To evaluate the part status, we compare different versions of our model and analyse the effects of different components. Specifically, we compute the part status detection rates, that is, a part status is correct if and only if its position and status (e.g., *open, close*) are both correct.

On the right of Fig. 6, we show the quantitative results of

4 versions of our ST-AOG. Here, we use HOG features for simplicity. "ST-AOG-ver[1]" refers to the ST-AOG without both view and part status penalty in Eqn. (8), "ST-AOG-ver[2]" refers to the ST-AOG without view penalty in Eqn. (8), and "ST-AOG" is the full model with all penalty in Eqn. (8). To investigate the effect of motion flow, we also compare the ST-AOG without motion flow, i.e., the "S-AOG". As can be seen, the view penalty is the most important factor, then is the part status and motion flow, and part status seems to have more importance. Interestingly, for some parts (e.g., "RB-Door"), adding more penalty or temporal info will decrease the performance.

Fig. 7 shows some qualitative running examples of ST-AOG. The first two rows show the successful results, the last row shows the failure examples. For better visualization, we cropped the image to just contain the interested car. We can see our model can localize parts and recognize their statuses fairly good with different viewpoints and car types, but may report wrong results when people occlude the car, the viewpoints are mis-detected, or there are background clutters.

7. Conclusion

This paper proposed a ST-AOG to recognize car fluents at semantic part-level from video. The ST-AOG integrates the motion features in temporal, and a LBP-based DP method can be used in inference. The model parameters are learned under the latent structural SVM (LSSVM) framework. To promote the research of fluents, we have collected a new Car-Fluent dataset with detailed annotations of viewpoints, car types, part bounding boxes, and part statuses. This dataset presents new challenges to computer vision, and complements existing benchmarks. Our experiments verified the ability of proposed ST-AOG on fluent recognition, part localization, and part status estimation.

In future work, we will integrate human and car jointly, and study human-car interactions to help understand human actions/intents based on car fluent recognition.

References

[1] M. Andriluka, S. Roth, and B. Schiele. Pictorial structures revisited: People detection and articulated pose estimation. In *CVPR*, 2009.

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

[3] M. J. Black, D. J. Fleet, and Y. Yacoob. A framework for modeling appearance change in image sequences. In *ICCV*, pages 660–667, 1998.

[4] M. J. Black, D. J. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. *Computer Vision and Image Understanding*, 78(1):8 – 31, 2000.

[5] A. Blum and M. L. Furst. Fast planning through planning graph analysis. In *IJCAI*, 1995.

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

[7] M. Brand. The "inverse hollywood problem": From video to scripts and storyboards via causal analysis. In *In Proceedings, AAAI97*, 1997.

[8] X. Chen, R. Mottaghi, X. Liu, S. Fidler, R. Urtasun, and A. Yuille. Detect what you can: Detecting and representing objects using holistic models and body parts. In *CVPR*, 2014.

[9] X. Chen and A. Yuille. Articulated pose estimation by a graphical model with image dependent pairwise relations. In *NIPS*, 2014.

[10] Y. Chen, L. Zhu, C. Lin, A. L. Yuille, and H. Zhang. Rapid inference on a novel and/or graph for object detection, segmentation and parsing. In *NIPS*, 2007.

[11] A. Cherian, J. Mairal, K. Alahari, and C. Schmid. Mixing body-part sequences for human pose estimation. In *CVPR*, 2014.

[12] J.-Y. Choi, K.-S. Sung, and Y.-K. Yang. Multiple Vehicles Detection and Tracking based on Scale-Invariant Feature Transform. In *ITSC*, 2007.

[13] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[14] C. Desai and D. Ramanan. Detecting actions, poses, and objects with relational phraselets. In *ECCV*, 2012.

[15] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[16] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010.

[17] P. F. Felzenszwalb and D. P. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 2012.

[18] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. In *CVPR*, 2009.

[19] A. Fire and S.-C. Zhu. Using causal induction in humans to learn and infer causality from video. In *CogSci*, 2013.

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

[21] R. Girshick, P. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *NIPS*, 2011.

[22] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015.

[23] A. Gupta, A. Kembhavi, and L. S. Davis. Observing human-object interactions: Using spatial and functional compatibility for recognition. *TPAMI*, 2009.

[24] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. In *CVPR*, 2009.

[25] K. He, L. Sigal, and S. Sclaroff. Parameterizing object detectors in the continuous pose space. In *ECCV*, 2014.

[26] D. Heckerman. A bayesian approach to learning causal networks. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995.

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

[28] S. Hongeng, R. Nevatia, and F. Brmond. Video-based event recognition: activity representation and probabilistic recognition methods. *Computer Vision and Image Understanding*, 96(2):129–162, 2004.

[29] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[30] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[31] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1:541–551, 1989.

[33] J. T. Lee, M. S. Ryoo, and J. K. Aggarwal. View independent recognition of human-vehicle interactions using 3-d models. In *WMVC*, 2009.

[34] T. S. H. Lee, S. Fidler, and S. Dickinson. Detecting curved symmetric parts using a deformable disc model. In *ICCV*, 2013.

[35] M. J. Leotta and J. L. Mundy. Vehicle Surveillance with a Generic, Adaptive, 3D Vehicle Model. *TPAMI*, 2011.

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

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

[38] L.-J. Li and F.-F. Li. What, where and who? classifying events by scene and object recognition. In *ICCV*, 2007.

[39] E. T. Mueller. *Commonsense Reasoning*. Morgan Kaufmann, San Francisco, 2006.

[40] I. Newton and J. Colson. *The Method of Fluxions And Infinite Series: With Its Application to the Geometry of Curve-Lines*. Nourse, 1736.

[41] B. X. Nie, C. Xiong, and S.-C. Zhu. Joint action recognition and pose estimation from video. In *CVPR*, 2015.

[42] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. Davis,

E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai. A large-scale benchmark dataset for event recognition in surveillance video. In *CVPR*, 2011.

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

[44] D. Park and D. Ramanan. N-best maximal decoders for part models. In *ICCV*. IEEE, 2011.

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

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

[47] R. J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: A systematic survey. *Trans. Img. Proc.*, 14(3):294–307, Mar. 2005.

[48] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In *NIPS*, 2015.

[49] E. Rivlin, S. J. Dickinson, and A. Rosenfeld. Recognition by functional parts. In *CVPR*, 1994.

[50] M. S. Ryoo, J. T. Lee, and J. K. Aggarwal. Video scene analysis of interactions between humans and vehicles using event context. In *CIVR*. ACM, 2010.

[51] M. A. Sadeghi and A. Farhadi. Recognition using visual phrases. In *CVPR*, 2011.

[52] J. Sanchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 2013.

[53] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller. Fine-grained categorization for 3d scene understanding. In *BMVC*, 2012.

[54] J. Tao, M. Enzweiler, U. Franke, D. Pfeiffer, and R. Klette. What is in front? multiple-object detection and tracking with dynamic occlusion handling. In G. Azzopardi and N. Petkov, editors, *CAIP*, 2015.

[55] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*. IEEE, 2015.

[56] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *CVPR*, 2015.

[57] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *IJCV*, 2013.

[58] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Action Recognition by Dense Trajectories. In *CVPR*, 2011.

[59] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[60] J. Wang and Y. Wu. Learning maximum margin temporal warping for action recognition. In *ICCV*, 2013.

[61] D. Weinland, R. Ronfard, and E. Boyer. A survey of vision-based methods for action representation, segmentation and recognition. *Computer Vision and Image Understanding*, 115(2):224–241, 2011.

[62] Y. Weiss. Correctness of local probability propagation in graphical models with loops, 2000.

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

[64] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. In *CVPR*, 2015.

[65] L. Yang, P. Luo, C. C. Loy, and X. Tang. A large-scale car dataset for fine-grained categorization and verification. In *CVPR*, 2015.

[66] Y. Yang, Y. Li, C. Fermuller, and Y. Aloimonos. Robot learning manipulation action plans by Watching unconstrained videos from world wide web. In *AAAI 2015*, Austin, US, jan 2015.

[67] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, 2011.

[68] B. Yao and L. Fei-Fei. Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses. *TPAMI*, 2012.

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

[70] L. Zhu, Y. Chen, A. L. Yuille, and W. T. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010.

[71] S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2006.

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

[73] S. Zuffi, J. Romero, C. Schmid, and M. J. Black. Estimating human pose with flowing puppets. In *ICCV*, 2013.