Online Object Tracking, Learning and Parsing with And-Or Graphs

Yang Lu, Tianfu Wu* and Song-Chun Zhu
Center for Vision, Cognition, Learning and Art
Department of Statistics, University of California, Los Angeles, USA
yanglv@ucla.edu, {tfwu, sczhu}@stat.ucla.edu

Abstract

This paper presents a framework for simultaneously tracking, learning and parsing objects with a hierarchical and compositional And-Or graph (AOG) representation. The AOG is discriminatively learned online to account for the appearance (e.g., lighting and partial occlusion) and structural (e.g., different poses and viewpoints) variations of the object itself, as well as the distractors (e.g., similar objects) in the scene background. In tracking, the state of the object (i.e., bounding box) is inferred by parsing with the current AOG using a spatial-temporal dynamic programming (DP) algorithm. When the AOG grows big for handling objects with large variations in long-term tracking, we propose a bottom-up/top-down scheduling scheme for efficient inference, which performs focused inference with the most stable and discriminative small sub-AOG. During online learning, the AOG is re-learned iteratively with two steps: (i) Identifying the false positives and false negatives of the current AOG in a new frame by exploiting the spatial and temporal constraints observed in the trajectory; (ii) Updating the structure of the AOG, and re-estimating the parameters based on the augmented training dataset. In experiments, the proposed method outperforms state-of-the-art tracking algorithms on a recent public tracking benchmark with 50 testing videos and 30 publicly available trackers evaluated [34].

1. Introduction

1.1. Objective and Motivation

Given a specified object in the first frame of a video, the objective of online object tracking is to locate it in the subsequent frames with bounding boxes. Online object tracking, especially *long-term tracking*, is a very challenging problem due to (i) the variations of the object itself, which include the appearance and structural variations, scale changes, occlusions (partial or complete), and the situations of disappearing and reappearing, etc., and (ii) the complexity of the scene, which includes camera motion, the background clutter, distractors, illumination changes, and frame cropping, etc. In recent literature, object tracking has received much attention due to practical applications in video surveillance, activity and event prediction, etc.

This paper presents an online Tracking-Learning-Parsing (TLP) framework to address the issues above. The object to be tracked is modeled by an And-Or graph (AOG) [39] which is a hierarchical and compositional representation with a directed acyclic graph (DAG) structure. The hypothesis space of AOG is constructed using the quantization method [31] recently proposed for learning object detectors in PASCAL VOC datasets. The AOG is discriminatively trained online to account for the variations of objects against background stated above. Our TLP framework is of similar spirit to tracking-learning-detection (TLD) [19], tracking-by-detection [2] and self-paced learning of tracking [16]. Note that the AOG in this paper is learned for the tracked object instance, rather than at the object category level. The motivation of introducing the AOG is four-fold.

i) More representational power: Unlike TLD [19] and many others which model an object as a single template or a mixture of small number of templates and thus does not perform well for articulated objects (e.g., person), an AOG represents an object in a hierarchical and compositional manner which has three types of nodes: an And-node represents the rule of decomposing a complex structure (e.g., a walking person or a running basketball player) into simple ones; an Or-node represents alternative structures at both object and part levels which can capture different poses and viewpoints and partial occlusion; and a Terminal-node grounds the representational symbol to image data using different appearance templates to capture local appearance change. Both the structure and appearance of the AOG will be discriminatively trained online to account for the variations of a tracked object against its scene backgrounds.

ii) More flexible computing schemes: Firstly, due to the DAG structure of an AOG, we can use a dynamic programming (DP) algorithm in inference; Secondly, the compositional property embedded in an AOG naturally leads to different bottom-up/top-down computing schemes as the α-β-γ computing processes studied in [32], which can track

*Tianfu Wu is the corresponding author

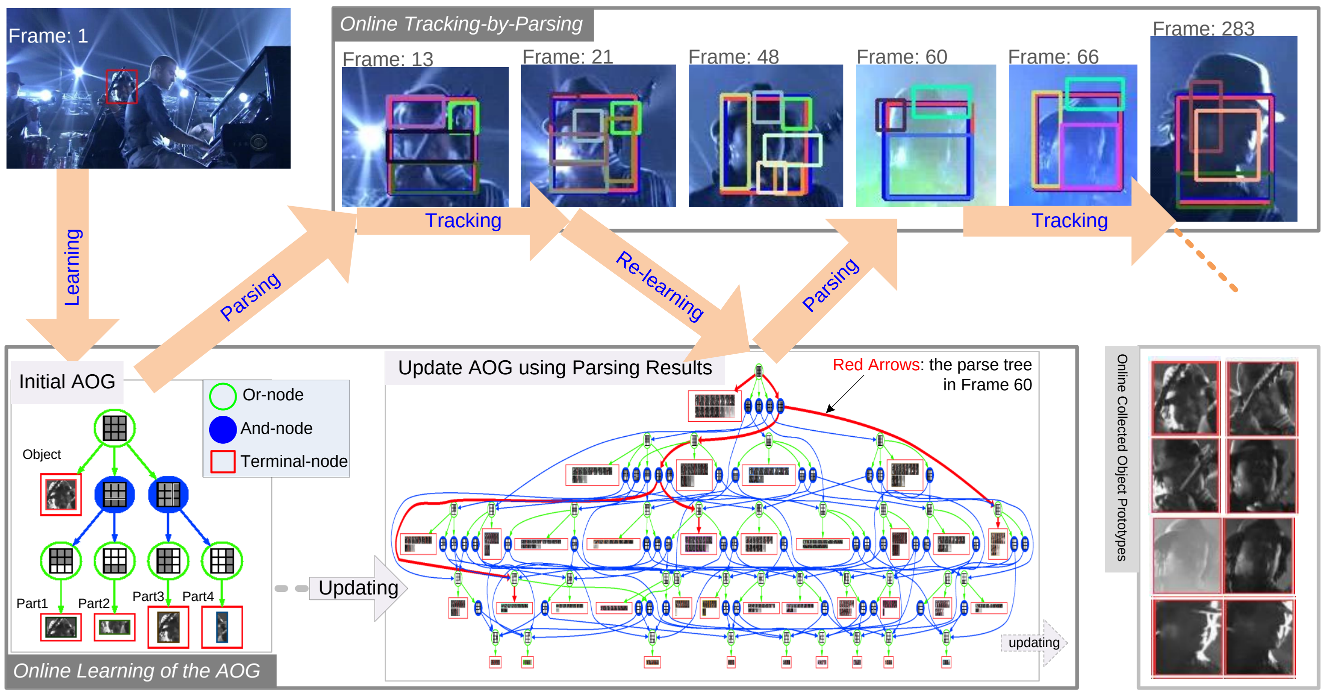

Figure 1. Illustration of our tracking-learning-parsing (TLP) framework using the "shaking" video in the dataset [34]. See text for details. Also see the video demos in the supplementary material. (Best viewed in color and magnification)

the object by matching the object template directly (α), or computing some discriminative parts first and then combine them into object (β), or doing both ($\alpha + \beta$). Usually, at the beginning of tracking, the AOG has a few number of nodes so we can track the object by parsing with the AOG using DP efficiently. As time evolves, the AOG can grow through online learning, especially for objects with large variations in long-term tracking, and thus faster inference is entailed for the sake of real time applications.

iii) More robust tracking and online learning strategies: While the whole object has large variations or is partially occluded from time to time during tracking, some parts might remain stable and are less likely to be occluded, so they can be used to robustly track the object using a bottom-up/top-down computing scheme, and also to improve the accuracy of appearance adaptation of Terminal-nodes (e.g., identifying appearance changes or occlusion of other nodes). This idea is similar to finding good features to track [30], and we find good part(s) online for both tracking and learning.

iv) Fine-grained tracking results: Beside predicting the bounding boxes of the tracked object, the output of our method (i.e., the parse trees) has much more information which is potentially useful for other modules beyond the tracking such as activity or event prediction.

1.2. Method Overview

As illustrated in Fig.1, our framework consists of three components:

i) The AOG for modeling the tracked object. Given the input bounding box of the object in the first frame (top-left), we divide the bounding box into a $r \times c$ cells (3×3 here). The set of primitive parts are then enumerated in the $r \times c$ cells, which quantize the hypothesis space of AOG using the method proposed in [31]. The quantization is capable of exploring a large number of latent part configurations (capturing discriminative and stable parts at different frames), meanwhile it makes the problem of online learning AOG feasible. We will elaborate the construction of the AOG in Sec. 3.1. See two examples, the initial AOG and one online updated AOG , in the bottom-left of Fig. 1.

ii) The spatial-temporal DP algorithm for tracking-by-parsing with AOG. At time t, given the previous bounding box, the spatial DP algorithm is used to compute the matching score and parse tree of a sliding window inside the search range (which can be either some neighborhood around the location and scale predicted based on the previous bounding box or the whole feature pyramid). A parse tree is a realization of the AOG by selecting the best child node for each encountered Or-node, and is the best interpretation of the object at current frame. For example, the parse tree of the object in Frame 60 is illustrated by the red arrows in the AOG (bottom-middle). We maintain a DP table memoizing the candidate object states generated by the spatial DP algorithm in the past n frames (e.g., 20 in our experiments). The temporal DP algorithm is then used to find the optimal solutions for the n frames, which can help correct tracking errors (i.e., false negatives and false positives collected online) by leveraging more spatial-temporal information. The formulation of the spatial-temporal DP is given in Sec. 4. We study the scheduling scheme for inference with a big AOG in handling objects with large variations during tracking (Sec. 5.3).

iii) The online learning of the AOG. The AOG is discriminatively trained online with only the whole object bounding box in the first frame being given. In this paper, we adopt the weakly-labeled latent SVM framework [12]. Details on online learning is given in Sec.5. The learning is done incrementally as time evolves, starting with a small set of positive examples (bootstrapped based on the given bounding box) and a set of negative examples (mined from outside of the given bounding box) to train the initial AOG. In the subsequent frames, the AOG is re-learned iteratively with two steps: The first step collects the false positives and false negatives of the current AOG in a new frame by exploring the temporal and spatial constraints in the trajectory, similar to the P-N learning proposed in TLD [19], and the second step updates the structure of the AOG (e.g., adding a new object template and/or some part configurations and corresponding templates) if necessary, and re-estimates the parameters based on the augmented training dataset. *One key issue of learning the AOG online* is how to maintain the purity of the positive and negative training set collected online (similar issue of learning single object template is discussed in [16] and [19]). The spatial-temporal DP algorithm for tracking-by-parsing stated above can help maintain the robustness of learning. Some object examples collected online are shown in the bottom-right in Fig. 1.

In experiments, the proposed framework is tested on a recent public benchmark [34] consisting of 50 video clips with different types of variations. Experimental results show that our method outperforms the state-of-the-art tracking methods consistently.

2. Related Work

In the literature of object tracking [36], either single object tracking or multiple-object tracking, there are often two types of settings:

i) The offline visual tracking [37, 28, 6], which assumes the whole video sequence has been recorded already, and then utilizes two steps; the first step generates object proposals in all frames of the input video by using some offline trained detectors (such as DPMs [11]) and then obtains "tracklets", and the second step finds the optimal object trajectory (or trajectories for multiple objects) by solving an optimization problem (e.g., the K-shortest path or min-cost flow formulation) for the data association.

ii) The online visual tracking, which is designed for live videos, and starts tracking when the bounding box of an object of interest was specified in certain frame. Most popular methods can be divided into four streams: (1) Appearance modeling of object itself as a whole, such as the incremental learning [29], kernel-based [8], particle filtering [17], sparse coding [26] and 3D-DCT representation [23]. (2) Appearance modeling of object with parts, such as patch-based [22], coupled 2-layer model [7] and adaptive sparse appearance [18]. The major limitation of the appearance modeling of object itself is the lack of background models, especially when there are distracotrs (e.g., players in sport games). To address this issue, it leads to so called discriminant tracking. (3) Tracking by discrimination using a single classifier, such as the support vector tracking [3], multiple instance learning [4], the struck [14], the circulant structure-based kernel method [15], and the discriminant saliency based tracking [25]. (4) Tracking by part-based discriminative models, such as the online extension of the DPM model [35], and the structure preserving tracking method [38].

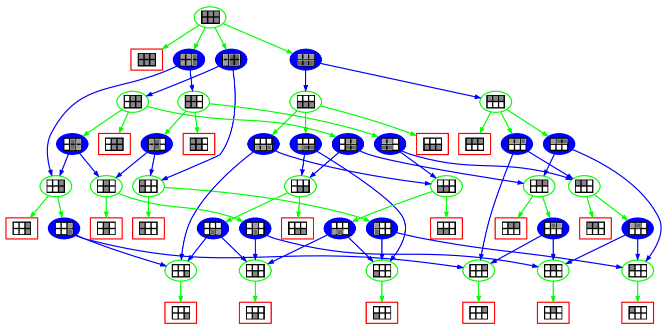
Figure 2. Illustration of constructing the compositional space of AOG using the method in [31]. Here, the full structure of the AOG is constructed for a 2×3 grid.

Our method belongs to the fourth stream of online visual tracking. Unlike the predefined or fixed part configurations with the star-model structure used in previous work, we learn both the structure and appearance of the AOG online which is, to our knowledge, the first method to address the problem of online explicit structure learning in tracking. **Our contributions.** This paper makes four contributions to the long-term online visual tracking problem:

i) It presents a generic tracking-learning-parsing (TLP) framework which can learn and track objects with online discriminatively trained AOGs.

ii) It presents a spatial-temporal DP algorithm for tracking-by-parsing with AOG and outputs fine-grained tracking results using parse trees.

iii) It proposes a simple yet effective bottom-up/top-down scheduling scheme for inference when the AOG grows big in tracking.

iv) It outperforms the state-of-the-art tracking methods on a recent public benchmark [34].

3. Problem Formulation

3.1. The AOG and Structure Quantization

Let B denote the input bounding box. We first divide it into a $r \times c$-cell grid (e.g., 2×3 in Fig. 2). The maximum grid size is 3×3 cells in this paper to control the model complexity. The full structure of AOG is constructed using breadth-first search (BFS) [31]. By "full structure", it means all the possible compositions on top of the cell grid with binary composition being used for And-nodes.

The AOG is a directed acyclic graph, denoted by $\mathcal{G} = (V, E)$. The node set V consists of three subsets of Or-nodes, And-nodes and Terminal-nodes respectively, which represent different aspects of modeling objects in a grammatical manner [39]. From the top to bottom, the AOG consists of: *The object Or-node (plotted by green circles)*, which represents alternative object configurations; *A set of And-nodes (solid blue circles)*, each of which represents a typical configuration of the tracked object; *A set of part Or-nodes*, which handle local variations and configurations in a recursive manner; *A set of Terminal-nodes (red rectangles)*, which link the whole object and parts to the image data (i.e., grounding the symbols), and take into account appearance Or-node (i.e., local appearance mixture) and occlusions (e.g., the head-shoulder of a walking person before and after opening a sun umbrella). Note that some part Or-nodes are shared between different And-nodes.

A parse tree is an instantiation of the AOG with the best child node of each encountered Or-node being selected. See the example illustrated by the red arrows in Fig. 1. All the terminal-nodes in a parse tree represents a part configuration when collapsed to image domain, as the parsing results (rectangles in different colors) shown in top-right in Fig. 1.

3.2. Formulation of Object Tracking

Let Λ denote the image lattice on which the video frames are defined. Denote a sequence of video frames within time range $[0, T]$ by $I_{0:T} = \{I_0, \cdots, I_T\}$. Let $\mathcal{C}_t = (B_t, B_t^{(1)}, \cdots, B_t^{k_t})$ be the configuration collapsed from the parse tree of the tracked object in I_t where B_t is the object bounding box B_t and $(B_t^{(1)}, \cdots, B_t^{(k_t)})$ are a small number k_t of part bounding boxes within B_t.

The objective of tracking is to predict B_t in I_t, and we treat $(B_t^{(1)}, \cdots, B_t^{k_t})$ as latent variables which are modeled to leverage more information for computing B_t. Note that we do not track $(B_t^{(1)}, \cdots, B_t^{k_t})$ explicitly.

We first derive the formulation from the generative perspective by considering a first-order Hidden Markov Model as usual,

$$\text{The prior model:} \quad B_0 \sim p(B_0)\,, \tag{1}$$

$$\text{The motion model:} \quad B_t|B_{t-1} \sim p(B_t|B_{t-1})\,, \tag{2}$$

$$\text{The likelihood:} \quad I_t|B_t \sim p(I_t|B_t). \tag{3}$$

Instead of following the traditional derivation based on the prediction model $p(B_t|I_{0:t-1}) = \int p(B_t|B_{t-1})p(B_{t-1}|I_{0:t-1})dB_{t-1}$ and the updating model $p(B_t|I_{0:t}) = p(I_t|B_t)p(B_t|I_{0:t-1})/p(I_t|I_{0:t-1})$ (which is a marginal posterior probability), we seek to maximize a joint posterior probability directly,

$$\begin{aligned} p(B_{0:t}|I_{0:t}) &= p(B_{0:t-1}|I_{0:t-1})\frac{p(B_t|B_{t-1})p(I_t|B_t)}{p(I_t|I_{0:t-1})} \\ &= p(B_0|I_0)\prod_{i=1}^{t}\frac{p(B_i|B_{i-1})p(I_i|B_i)}{p(I_i|I_{0:i-1})}. \end{aligned} \tag{4}$$

By taking logarithm at both sides in Eqn.(4), we have,

$$\begin{aligned} B^*_{0:t} &= \arg\max_{B_{0:t}} \log p(B_{0:t}|I_{0:t}) \\ &= \arg\max_{B_{0:t}} \{\log p(B_0) + \log p(I_0|B_0) + \\ &\quad \sum_{i=1}^{t}[\log p(B_i|B_{i-1}) + \log p(I_i|B_i)]\}. \end{aligned} \tag{5}$$

where the image data term $p(I_0)$ and $\sum_{i=1}^{t} p(I_i|I_{0:i-1})$ are not included in the maximization as they are treated as constant terms. In online tracking, we have groundtruth for B_0 and thus $p(I_0|B_0)$ can also be treated as known after the object model is trained based on B_0. Then, Eqn.(5) can be reproduced as,

$$\begin{aligned} B^*_{1:t} &= \arg\max_{B_{1:t}} \log p(B_{1:t}|I_{0:t}, B_0) \\ &= \arg\max_{B_{1:t}} \{\sum_{i=1}^{t}[\log p(B_i|B_{i-1}) + \log p(I_i|B_i)]\}. \end{aligned} \tag{6}$$

which leads to the spatial-temporal DP algorithm for tracking-by-parsing with AOG.

4. Tracking-by-Parsing with AOG using Spatial-Temporal DP Algorithm

By following the derivation in [32], we show that only the log-likelihood ratio matters in computing the log-likelihood $\log p(I_i|B_i)$. We can obtain,

$$\begin{aligned} p(I_i|B_i) &= p(I_{\Lambda_{B_i}}, I_{\overline{\Lambda_{B_i}}}|B_i) = p(I_{\Lambda_{B_i}}|B_i)q(I_{\overline{\Lambda_{B_i}}}) \\ &= q(I_\Lambda)\frac{p(I_{\Lambda_{B_i}}|B_i)}{q(I_{\Lambda_{B_i}})}, \end{aligned} \tag{7}$$

where $\overline{\Lambda_{B_i}}$ is the remaining domain (i.e., $\overline{\Lambda_{B_i}} \cup \Lambda_{B_i} = \Lambda$ and $\overline{\Lambda_{B_i}} \cap \Lambda_{B_i} = \emptyset$), and $q(I_\Lambda)$ is the probability model of scene background which does not need to be specified explicitly in the computation. This derivation gives an alternative explanation for the discriminant tracking *v.s.* tracking by appearance modeling of object itself.

So, we will treat Eqn.(6) from the discriminative perspective, i.e., we do not compute $\log p(I_i|B_i)$ and $\log p(B_i|B_{i-1})$ in the probabilistic way, instead we compute the matching scores of online discriminatively trained AOG. Denote by $\text{Score}(I_i|B_i) = \log \frac{p(I_{\Lambda_{B_i}}|B_i)}{q(I_{\Lambda_{B_i}})}$. We can

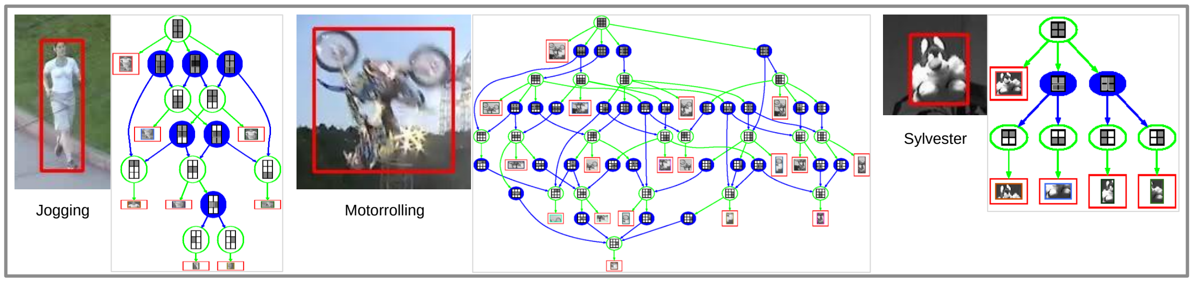

Figure 3. Examples of the initial AOGs learned for the "Jogging", "Motorrolling" and "Sylvester" in the benchmark [34]. Depending on both the object and the background, the initial AOGs for different objects have different complexity.

re-write Eqn.(6) in the minimization form,

$$B^*_{1:t} = \arg\min_{B_{1:t}} \text{Cost}(B_{1:t}|I_{0:t}, B_0; \mathcal{G}) \quad (8)$$
$$= \arg\min_{B_{1:t}} \{\sum_{i=1}^{t}[\text{Cost}(B_i|B_{i-1}) - \text{Score}(I_i|B_i; \mathcal{G})]\}.$$

where the scores are computed based on the online learned AOG so we add $\mathcal{G}$ to the equation.

The spatial DP computes $\text{Score}(I_i|B_i; \mathcal{G})$, and the temporal DP solves the optimal solution of $B^*_{1:t}$ in Eqn.(8).

4.1. The Spatial DP Algorithm

To compute $\text{Score}(I_i|B_i; \mathcal{G})$, we do parsing inside Λ_{B_i} with the current AOG $\mathcal{G}$ with the optimal configuration $\mathcal{C}^*_i$ being sought. We denote this parsing process by $\text{Parse}(I_i|B_i; \mathcal{G})$ which is given in Algorithm.1 in the supplementary material. The basic idea is that for a given candidate B_i, we want to find the best of all possible parse trees in the AOG, and for each parse tree we want to find the best part configuration (through local deformation of the Terminal-nodes). The DP algorithm is used to solve these two rounds of maximization efficiently.

In practice, at time t, given the previous bounding box B_{t-1}, the spatial search space is defined by the feature pyramid which is processed in a "center-surround" manner: the "center" means the neighborhood of both pyramid levels and corresponding spatial domain defined by B_{t-1}, and the "surround" is the remaining portion in the feature pyramid. We first run the spatial DP algorithm with the current AOG inside the "center". If the DP solution has high confidence matching score based on the online learned threshold, it will be accepted. Otherwise, it keeps all the candidates with scores greater than some threshold (e.g., 70% of the high confidence threshold), and then run DP algorithm in the "surround" with all the candidates kept in the similar manner, followed by running the temporal DP algorithm.

4.2. The Temporal DP Algorithm

Assume that all the candidates for $B_1, \cdots, B_t$ are memoized after running the spatial DP algorithm for tracking-by-parsing in I_1 to I_t, Eqn.(8) corresponds to the classic DP formulation with $-\text{Score}(I_i|B_i; \mathcal{G})$ being the local cost term and $\text{Cost}(B_i|B_{i-1})$ the pairwise cost term.

To compute $\text{Cost}(B_i|B_{i-1})$, we use a thresholded motion model, as experimented in [16]: the cost is 0 if the transition is accepted by the measured median flow [19] (which is a forward-backward extension of the Lucas-Kanade optimal flow [5]) and $+\infty$ otherwise.

In practice, we often do not need to run the temporal DP in the whole time range $[1, t]$, especially for long-term tracking, since the tracked object might have changed significantly, instead we only focus on some short time range (e.g., the past 20 frames used in our experiments).

5. Online Learning of the AOG

In this section, we present the online learning of AOG consisting of two components: (i) Learning the initial AOG given the the input bounding box in the first frame. (ii) Updating the AOG with the results from tracking-by-parsing.

Appearance feature and learning framework. We use the modified HOG feature used in DPM [11], and adopt the weakly-labeled latent SVM framework (WLLSVM) [12] to estimate the appearance parameters.

Local deformation. In this paper, we do not use the quadratic deformation term as done in the DPM, instead we use local max when summing the scores over child nodes for an And-node (as written in Algorithm.1 in the supplementary). The local deformation range is proportional to the side lengths of a terminal-node (e.g., 0.1 in this paper).

Denote by D^+_t the online collected positive dataset, and by D^-_t the online collected negative dataset at time t.

5.1. Learning the Initial AOG

We have $D^+_0 = \{(I_0, B_0)\}$. We augment it by warping a small number (20 in our experiments) of positives (i.e., creating new positives by adding random Gaussian noise and random small affine transformations). The initial D^-_0 use the whole remaining image $I_{\overline{\Lambda_{B_0}}}$ for mining hard negatives during training. This mining step improves the tracking significantly which is also observed in [16].

The initial AOG, denoted by $\mathcal{G}_0$. The structure is learned by pruning. We first train the full object template, denoted by ω. Then, the appearance parameters for each terminal-node v in the full AOG is initialized by cropping out the corresponding portion in ω, denoted by ω_v. We evaluate the "goodness" of a terminal-node v by its variance, over

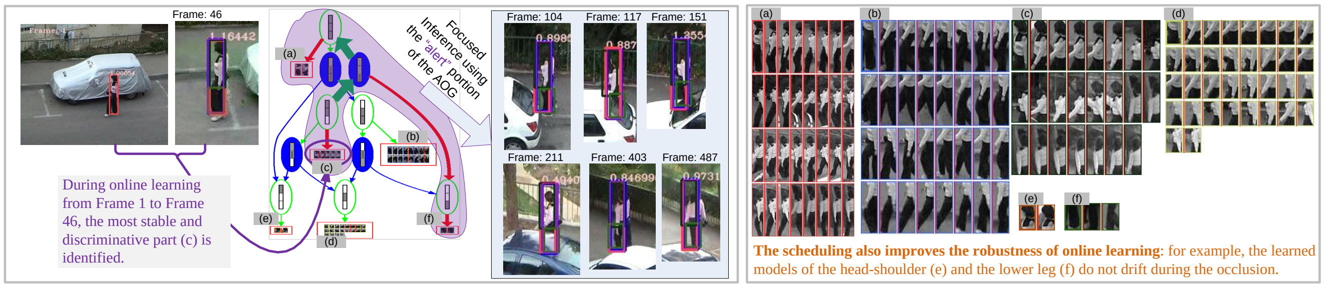
Frame: 46
During online learning from Frame 1 to Frame 46, the most stable and discriminative part (c) is identified.
(a)
(b)
(c)
(d)
(e)
(f)
Focused Inference using the "alert" portion of the AOG
Frame: 104
Frame: 117
Frame: 151
Frame: 211
Frame: 403
Frame: 487
The scheduling also improves the robustness of online learning: for example, the learned models of the head-shoulder (e) and the lower leg (f) do not drift during the occlusion.

Figure 4. Illustration of the scheduling, which improves both the computing efficiency and the robustness of online learning of the AOG.

all positives, of the full object template score minus the terminal-node score, which is used in the DPM cascade [10] to order the parts. The smaller the variance is, the more stable and important the terminal-node is. Then, we threshold the variance to prune the terminal nodes and divide the set of terminal-node based on the "on/off" states, i.e., $V_T = V_T^{on} \cup V_T^{off}$. Based on V_T^{on}, we learn the initial AOG. To maintain the structure, we require that all the child nodes of an And-node need to be turned "on", which is implemented by two rounds of traversal of the full AOG:

i) Turn on/off all the And-nodes and Or-nodes in the full AOG using DFS. An encountered Or-node turns on if one of its child turns on. An encountered And-node turns on if and only if all of its child nodes turn on.

ii) Retrieve the initial AOG using BFS. Start from the root Or-node, add all turned-on child nodes of an encountered Or-node or And-node, and add any encountered Terminal-node.

Note that some Terminal-nodes in V_T^{on} might not be included in the initial AOG, and they can be added back during updating if the condition was satisfied. Fig. 3 shows three examples of learned initial AOGs.

We re-train the appearance parameters of the whole initial AOG jointly using modified WLLSVM [12] and estimate the threshold for all nodes in the initial AOG using the PAA method [10]. After training, we keep the positive and negative support vectors in the training cache which will be reused in updating the AOG. With the learned initial AOG, we start tracking-by-parsing as stated in Sec.4. During the tracking, we update the AOG based on the tracking results.

5.2. Online Updating of the AOG

The goal of updating the AOG online is to account for both the structural and appearance variations of the tracked object, as well as to handle hard negatives (distractors) in the background.

We will keep the appearance parameters of terminal nodes in the initial AOG unchanged since they are learned with groundtruth input, which will help locate a re-appearing object (e.g., after moving out of the scene or being completely occluded). So, the appearance of a terminal-node is represented by a mixture (i.e., appearance Or-node) in updating.

At time t, with the tracking-by-parsing results, we update the AOG in the following way,

i) Maintaining D_t^+ and D_t^- based on D_{t-1}^+ and D_{t-1}^-: (1) If the tracking result B_t has a high confidence parsing score, it is added to D_t^+ and then all other high-scoring candidates generated during the search are added to D_t^-; (2) Correct the previously augmented examples in D_{t-1}^+ and D_{t-1}^- according to the consistency with the temporal DP result (i.e., correct previously false positives and false negatives). This controls the purity of the training dataset similar to the P-N learning in TLD [19].

ii) Updating the training cache based on the parse trees obtained from D_t^+ and D_t^- with the current model.

iii) Relearn the appearance parameters for all Terminal-nodes in the current AOG.

iv) Updating the structure of the AOG: (1) Initialize all the remaining Terminal-nodes based on the updated object template; (2) Evaluate the "goodness" of all the Terminal-nodes and turn them on/off as done in learning the initial AOG; (3) Retrieve the new structure of the AOG using DFS and BFS.

v) Retrain the AOG if the structure has been updated in step iv).

Please see the video demos in the supplementary material for illustrating the AOG learning and updating. We will study more theoretically-sound online learning framework for the AOG in tracking, which is a very challenge problem under general settings.

5.3. Scheduling in the AOG

When the tracked objects have large variations, the AOG can grow big, especially in long-term tracking. To improve the computational efficiency, we propose a simple yet effective scheduling scheme which do focused inference with identified "alert" sub-graph of the AOG (Fig. 4).

The scheduling is based on the ordering of nodes in the AOG and estimating a two-sided thresholds, i.e., early acceptance/early rejection threshold, for each node.

i) The Terminal-nodes in the AOG are sorted using the variance as "goodness" measure as done in the learning.

ii) A two-sided threshold is estimated for each Terminal-node and And-node using the decision policy method [33].

	AOGTracker	Struck[14]	CXT[9]	VTD[20]	VTS[21]	OAB[13]	CPF[27]	LSK[24]	Frag[1]	MIL[4]	SPO[38]
Prec.	**0.851**	0.773	0.658	0.650	0.645	0.604	0.599	0.589	0.582	0.574	0.587
Suc.	**0.748**	0.694	0.579	0.583	0.581	0.540	0.502	0.556	0.530	0.496	0.488

Table 1. Overall performance comparison of the top 10 trackers evaluated on the 50-video benchmark [34]. We follow the evaluation protocol proposed in [34] to compute the precision and success rate.

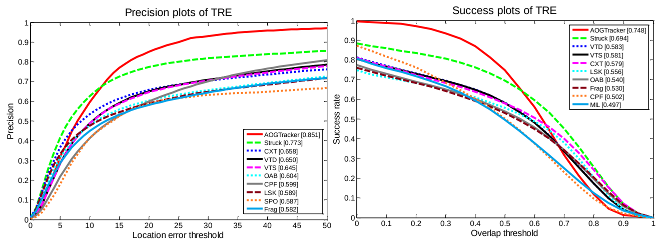

Figure 5. Plots of overall performance comparison for the 50 videos in the benchmark [34]. The proposed method ("AOG-Tracker") obtains better performance in terms of precision (left) and success (right) plot.

iii) The tracking-by-parsing (Algorithm.1 in the supplementary) is then modified: (1) Instead of following the DFS completely to compute scores in the step 0, we schedule the computation of the nodes according to the ordering; (2) Bottom-up computing with DFS in the AOG starting from the first Terminal-node in the ordering to prune the search space; and (3) Top-down verification using the two-sided threshold with the BFS in the AOG to exploit early stop.

By this scheduling, we can also improve the robustness of online learning through finding good part or partial part configuration to guide the updating of the AOG, especially for handling occlusion and large variations of some parts. See the illustration in Fig. 4.

With the scheduling scheme, our tracker can run 2 to 3 frames per second on a single CPU core.

6. Experiments

We test our method on a recent public benchmark [34] consisting of 50 video clips which have different challenging aspects such illumination variation, scale variation, non-rigid deformation, occlusion, and out-of-view, etc. For the benchmark, most published tracking algorithms[1] (30 publicly available trackers) are evaluated including Struck [14], IVT [29], MIL [4], TLD [19] and structure preserving tracker [38], etc.. We follow the same evaluation protocol proposed in [34]. Due to space limit, we show quantitative comparison results only in this section, and qualitative results and video demos showing the details of online learning of our method will be presented in the supplementary.

Overall, our method outperforms them consistently (see Fig.5 and Table. 1). In addition, Fig. 6 shows the comparison on different subsets such as non-rigid deformation and out-of-view subsets. Note that Fig. 5 and Fig. 6 show the top 10 trackers only for clarity.

[1]https://sites.google.com/site/trackerbenchmark/benchmarks/v10

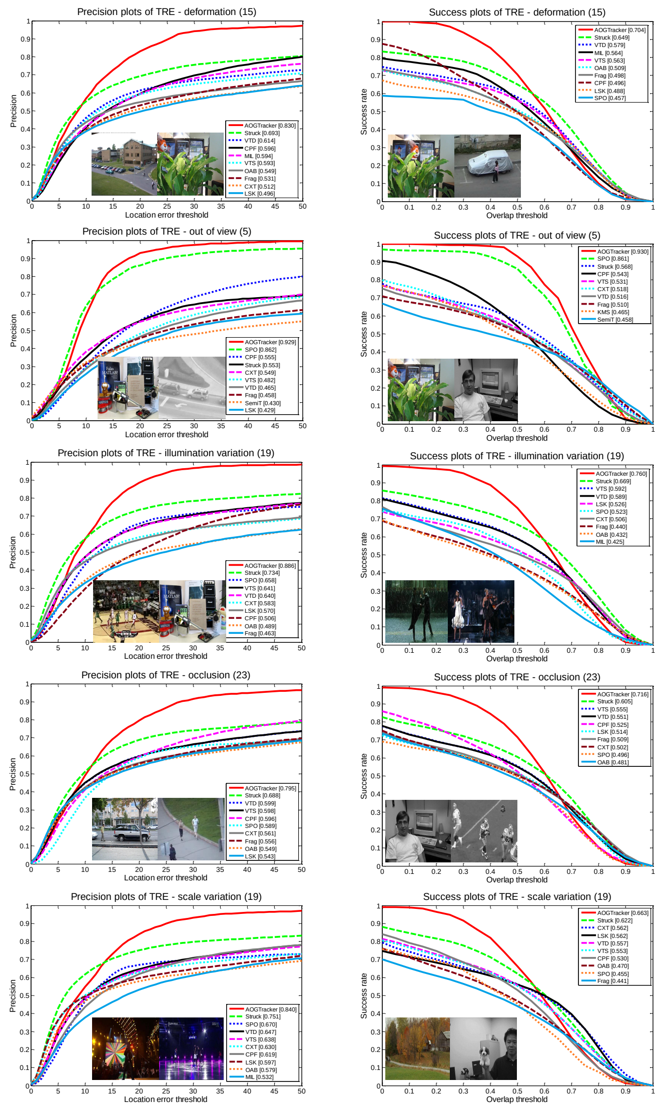

Figure 6. Detail comparisons in different subsets divided based on main variation of the object to be tracked (e.g., objects in 15 videos have the non-grid deformation including "Basketball", "Bolt", "Couple", "Crossing", etc.). The details of the subsets refer to [34]. The proposed method ("AOGTracker") obtains better or comparable performance in all the subsets.

For more close-view evaluation, we show four examples of the center distance error per frame in Fig. 7 with the top 4 tracker compared, which show that our method can handle occlusion, pose change and illumination well. The robustness of our AOG tracker lies in the hierarchical and compositional structure which are discriminatively trained online

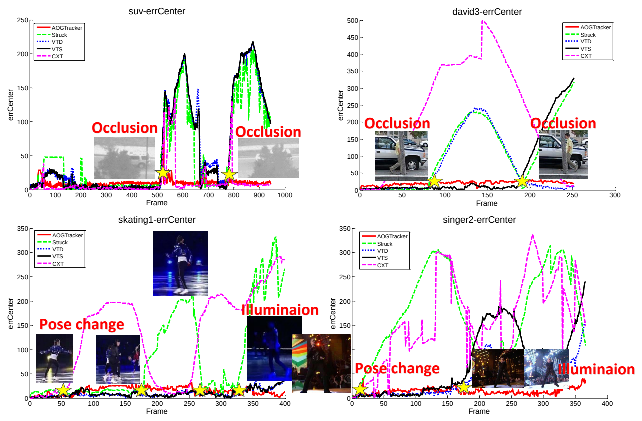

Figure 7. Comparisons on the center distance error per frame.

to account for the variations.

7. Discussion and Conclusion

This paper presents a tracking-learning-parsing framework for simultaneously tracking and learning objects with hierarchical models in the directed acyclic And-Or graph structure. We present a spatial-temporal dynamic programming algorithm for tracking-by-parsing with the AOG. We also present the method of online learning the AOG including its structure and appearance parameters. To handle the complexity when the AOG grows big in tracking objects with large variations, we study a simple yet effective scheduling scheme for inference, which improves both the computational efficiency and the robustness of learning. In experiments, we test our method on a recent public benchmark and experimental results show better performance.

Acknowledgment: This work is supported by DARPA MSEE project FA 8650-11-1-7149, MURI grant ONR N00014-10-1-0933, and NSF IIS1018751. We thank Dr. Brandon Rothrock for helpful discussion.

References

[1] A. Adam, E. Rivlin, and I. Shimshoni. Robust fragments-based tracking using the integral histogram. In *CVPR*, 2006.

[2] M. Andriluka, S. Roth, and B. Schiele. People-tracking-by-detection and people-detection-by-tracking. In *CVPR*, 2008.

[3] S. Avidan. Support vector tracking. *PAMI*, 26(8):1064–1072, 2004.

[4] B. Babenko, M.-H. Yang, and S. Belongie. Robust object tracking with online multiple instance learning. *PAMI*, 33(8):1619–1632, 2011.

[5] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221–255, 2004.

[6] J. Berclaz, F. Fleuret, E. Türetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *PAMI*, 33(9):1806–1819, 2011.

[7] L. Cehovin, M. Kristan, and A. Leonardis. Robust visual tracking using an adaptive coupled-layer visual model. *PAMI*, 35(4):941–953, 2013.

[8] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *PAMI*, 25(5):564–575, 2003.

[9] T. B. Dinh, N. Vo, and G. G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *CVPR*, 2011.

[10] P. Felzenszwalb, R. Girshick, and D. McAllester. Cascade object detection with deformable part models. In *CVPR*, 2010.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

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

[13] H. Grabner, M. Grabner, and H. Bischof. Real-time tracking via online boosting. In *BMVC*, 2006.

[14] S. Hare, A. Saffari, and P. H. S. Torr. Struck: Structured output tracking with kernels. In *ICCV*, 2011.

[15] J. Henriques, R. Caseiro, P. Martins, and J. Batista. Exploiting the circulant structure of tracking-by-detection with kernels. In *ECCV*, 2012.

[16] J. S. S. III and D. Ramanan. Self-paced learning for long-term tracking. In *CVPR*, 2013.

[17] M. Isard and A. Blake. Condensation - conditional density propagation for visual tracking. *IJCV*, 29(1):5–28, 1998.

[18] X. Jia, H. Lu, and M.-H. Yang. Visual tracking via adaptive structural local sparse appearance model. In *CVPR*, 2012.

[19] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. *PAMI*, 34(7):1409–1422, 2012.

[20] J. Kwon and K. M. Lee. Visual tracking decomposition. In *CVPR*, 2010.

[21] J. Kwon and K. M. Lee. Tracking by sampling trackers. In *ICCV*, 2011.

[22] J. Kwon and K. M. Lee. Highly nonrigid object tracking via patch-based dynamic appearance modeling. *PAMI*, 35(10):2427–2441, 2013.

[23] X. Li, A. R. Dick, C. Shen, A. van den Hengel, and H. Wang. Incremental learning of 3d-dct compact representations for robust visual tracking. *PAMI*, 35(4):863–881, 2013.

[24] B. Liu, J. Huang, L. Yang, and C. A. Kulikowski. Robust tracking using local sparse appearance model and k-selection. In *CVPR*, 2011.

[25] V. Mahadevan and N. Vasconcelos. Biologically inspired object tracking using center-surround saliency mechanisms. *PAMI*, 35(3):541–554, 2013.

[26] X. Mei and H. Ling. Robust visual tracking and vehicle classification via sparse representation. *PAMI*, 33(11):2259–2272, 2011.

[27] P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In *ECCV*, 2002.

[28] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR*, 2011.

[29] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang. Incremental learning for robust visual tracking. *IJCV*, 77(1-3):125–141, 2008.

[30] J. Shi and C. Tomasi. Good feature to track. In *CVPR*, 1994.

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

[32] T. Wu and S. C. Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *IJCV*, 93(2):226–252, 2011.

[33] T. Wu and S.-C. Zhu. Learning near-optimal cost-sensitive decision policy for object detection. In *ICCV*, 2013.

[34] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *CVPR*, 2013.

[35] R. Yao, Q. Shi, C. Shen, Y. Zhang, and A. van den Hengel. Part-based visual tracking with online latent structural learning. In *CVPR*, 2013.

[36] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.

[37] L. Zhang, Y. Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *CVPR*, 2008.

[38] L. Zhang and L. van der Maaten. Structure preserving object tracking. In *CVPR*, 2013.

[39] S. C. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362, 2006.