A Causal And-Or Graph Model for Visibility Fluent Reasoning in Tracking Interacting Objects

Yuanlu Xu[1]*, Lei Qin[2]*, Xiaobai Liu[3], Jianwen Xie[4], Song-Chun Zhu[1]
[1]University of California, Los Angeles [2]Inst. Computing Technology, Chinese Academy of Sciences
[3]Dept. Computer Science, San Diego State University [4]Hikvision Research Institute, USA
yuanluxu@cs.ucla.edu, qinlei@ict.ac.cn, xiaobai.liu@mail.sdsu.edu
jianwen.xie@hikvision.com, sczhu@stat.ucla.edu

Abstract

Tracking humans that are interacting with the other subjects or environment remains unsolved in visual tracking, because the visibility of the human of interests in videos is unknown and might vary over time. In particular, it is still difficult for state-of-the-art human trackers to recover complete human trajectories in crowded scenes with frequent human interactions. In this work, we consider the visibility status of a subject as a fluent variable, whose change is mostly attributed to the subject's interaction with the surrounding, e.g., crossing behind another object, entering a building, or getting into a vehicle, etc. We introduce a Causal And-Or Graph (C-AOG) to represent the causal-effect relations between an object's visibility fluent and its activities, and develop a probabilistic graph model to jointly reason the visibility fluent change (e.g., from visible to invisible) and track humans in videos. We formulate this joint task as an iterative search of a feasible causal graph structure that enables fast search algorithm, e.g., dynamic programming method. We apply the proposed method on challenging video sequences to evaluate its capabilities of estimating visibility fluent changes of subjects and tracking subjects of interests over time. Results with comparisons demonstrate that our method outperforms the alternative trackers and can recover complete trajectories of humans in complicated scenarios with frequent human interactions.

1. Introduction

Tracking objects of interest in videos is a fundamental computer vision problem that has great potentials in many video-based applications, e.g., security surveillance, disaster response, and border patrol. In these applications, a critical problem is how to obtain the complete trajectory of the object of interest while observing it moving in the scene through camera view. This is a challenging problem since an object of interest might undergo frequent interactions with the surrounding, e.g., entering a vehicle or a building, or with the other objects, e.g., passing behind another sub-

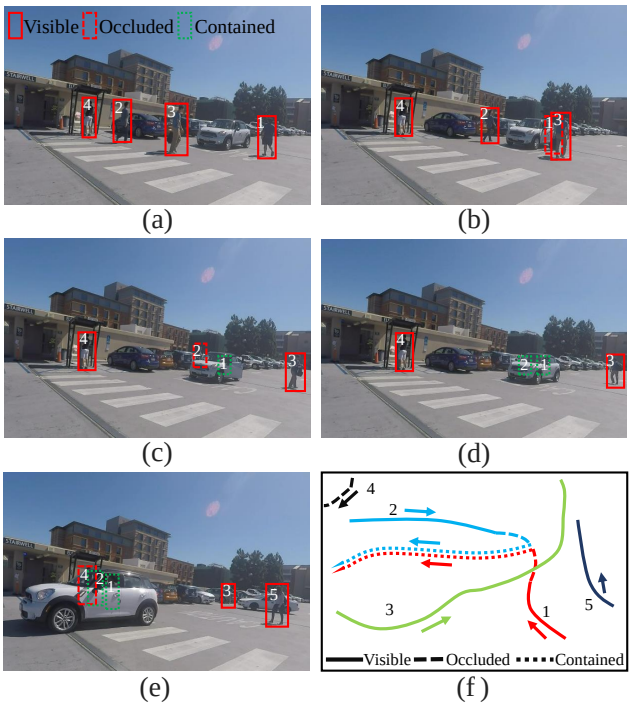

Figure 1. **Illustration of visibility fluent changes.** There are three states: visible, occluded, contained. When a person approaches a vehicle, its state changes from "visible" to "occluded" to "contained", such as the person$_1$ and person$_2$ (a-e). When a vehicle passes, the person$_4$ is occluded. The state of person$_4$ changes from "visible" to "occluded" in (d-e). (f) shows the corresponding top-view trajectories of different persons. The numbers are the persons' IDs. The arrows indicate the moving direction.

*Yuanlu Xu and Lei Qin contributed equally to this paper. This work is supported by ONR MURI Project N00014-16-1-2007, DARPA XAI Award N66001-17-2-4029, and NSF IIS 1423305, 1657600, and in part by National Natural Science Foundation of China: 61572465, 61390510, 61732007. The correspondence author is Xiaobai Liu.

ject. With these interactions, the visibility status of a subject will be varying over time, e.g., changing from "invisible" to "visible" and vice versa. In the literature, most state-of-the-art trackers utilize appearance or motion cues to localize subjects in video sequences and are likely to fail to track the subjects whose visibility status keep changing.

To deal with the above challenges, in this work, we propose to explicitly reason subjects' visibility status over time, while tracking the subjects of interests in surveillance videos. Traditional trackers are likely to fail when the target become invisible due to occlusion, our proposed method could jointly infer objects' locations and visibility fluent changes, thus helping to recover the complete trajectories. The proposed techniques, with slight modifications, can be generalized to other scenarios, e.g., hand-held cameras, driverless vehicles, etc.

The key idea of our method is to introduce a fluent variable for each subject of interest to explicitly indicate his/her visibility status in videos. Fluent was firstly used by Newton to denote the time varying status of an object. It is also used to represent the varying object status in commonsense reasoning [27]. In this paper, the visibility status of objects can be described as fluents varying over time. As illustrated in Fig. 1, the person_3 and person_5 are walking through the parking lot, while the person_1 and person_2 are entering a sedan. The visibility status of person_1's and person_2's changes first from "visible" to "occluded", and then to "contained". This group example demonstrates how objects' visibility fluents change over time along with their interactions to the surrounding.

We introduce a graphical model, i.e. Causal And-Or graph (C-AOG), to represent the causal relationships between object's activities (actions/sub-events) and object's visibility fluent changes. The visibility status of an object might be caused by multiple actions, and we need to reason the actual causality from videos. These actions are alternative choices that lead to the same occlusion status, and form the Or-nodes. Each leaf node indicates an action or sub-event that can be described by And-nodes. Taking the videos shown in Fig. 1 for instance, the status of "occluded" can be caused by the following actions: (i) walking behind a vehicle; (ii) walking behind a person; or (iii) inertial action that maintains the fluent unchanged.

The basic hypothesis of this model is that, for a particular scenario (e.g., parking-lot), there are only a limited number of actions that can cause the fluent to change. Given a video sequence, we need to create the optimal C-AOG and select the best choice for each Or-node in order to obtain the optimal causal parse graph, which is shown as red lines in Fig. 3(a).

We develop a probabilistic graph model to reason object's visibility fluent changes using C-AOG representation. Our formula integrates object tracking purposes as well to enable joint solution of tracking and fluent change reasoning, which are mutually beneficial. In particular, for each subject of interest, our method uses two variables to represent (i) subjects' positions in videos; and (ii) visibility status as well as the best causal parse graph. We utilize a Markov Chain Prior model to describe the transitions of these variables, i.e., the current state of a subject is only dependent on the previous state. We then reformulate the problem into an Integer Linear Programming model, and utilize dynamic programming to search the optimal states over time.

In experimental evaluations, the proposed method is tested on a set of challenging sequences that include frequent human-vehicle or human-human intersections. Results show that our method can readily predict the correct visibility status and recover the complete trajectories. In contrast, most of the alternative trackers can only recover part of the trajectories due to the occlusion or containment.

Contributions. There are three major contributions of the proposed framework: (i) a Causal And-Or Graph (C-AOG) model to represent object visibility fluents varying over time; (ii) a joint probabilistic formulation for object tracking and fluent reasoning; and (iii) a new occlusion reasoning dataset to cover objects with diverse fluent changes.

2. Related Work

The proposed research is closely related to the following three research streams in computer vision and AI.

Multiple object tracking has been extensively studied in the past decades. In the past literatures, tracking-by-detection has become the mainstream framework [37, 7, 40, 41, 9, 8]. Specifically, a general detector [11, 30] is first applied to generate detection proposals, and then data association techniques [4, 6, 42] are employed to link detection proposals over time in order to get object trajectories. Our approach also follows this pipeline, but is more focused on the reasoning of object visibility status.

Tracking interacting objects studies a more specific problem of tracking entangled objects. Some works [34, 35, 25] try to model the object appearing and disappearing phenomena globally, yielding strong assumptions on appearance, location or motion cues. On the contrary, other works attempt to model human-object and human-human interactions under specific scenarios, such as social activities [5, 31], team sports [24], and people carrying luggage [3]. In this paper, we propose a more principled way to track objects with both short-term interactions, e.g., passing behind another object, or long-term interactions, e.g., entering a vehicle and moving together.

Causal-effect reasoning is a popular topic in AI but has not received much attentions in the field of computer vision. It studies, for instances, the difference between co-occurrence and causality, and aims to learn causal knowledge automatically from low-level observations, e.g., im-

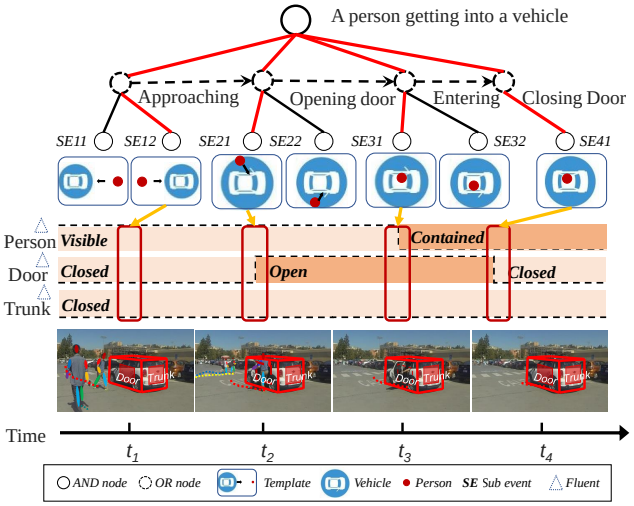

Figure 2. **Illustration of a person's actions and her visibility fluent changes** when she enters a vehicle.

ages or videos. There are two popular causality models: Bayesian Network [16, 28] and grammar models [17, 23]. Grammar models [39, 10, 33] are powerful tool for modeling high-level human knowledge in specific domains. Notably, Fire and Zhu [13] have introduced a causal grammar to infer causal-effect relationship between object's status, e.g., door open/close, and agent's actions, e.g., pushing the door. They studied this problem using manually designed rules and video sequences in lab settings. In this work, we extend the causal grammar models to infer objects' visibility fluent and ground the task on challenging videos in surveillance systems.

3. Representation

In this paper, we define three states for visibility fluent reasoning: **visible**, (partially/fully) **occluded**, and **contained**. Most multiple object tracking methods are based on tracking-by-detection framework, which obtain good performance in visible and partially occluded situations. However, when full occlusions take place, these trackers usually regard the disappearing-and-reappearing objects as new objects. Although objects in fully occluded and contained states are invisible, there are still evidences to infer the locations of objects and fill-in the complete trajectory. We can distinguish object being fully occluded and object being contained from three empirical observations.

Firstly, *motion independence*. In fully occluded state, such as a person staying behind a pillar, the motion of the person is independent of the pillar. While in contained state, such as a person sitting in a vehicle, or a bag in the trunk, the position and motion of the person/bag would be the same as the vehicle. Therefore, the inference of the visibility fluent of the object is important in tracking objects accurately in a complex environment.

Secondly, *coupling actions and object fluent changes*. For example, as illustrated in Fig. 2, if a person gets into a vehicle, the related sequential atomic actions are: approaching a vehicle, opening the vehicle door, getting into the vehicle, and closing the vehicle door; the related object fluent changes are vehicle door *closed* $\rightarrow$ *open* $\rightarrow$ *closed*. The fluent change is a consequence of agent actions. If the fluent-changing actions do not happen, the object should maintain its current fluent. For example, a person that is contained in a vehicle will remain contained unless he/she opens the vehicle door and gets out of the vehicle.

Thirdly, *visibility in the alternative camera views*. In full occlusion state, such as a person occluded by a pillar, though the person could not be observed from the current viewpoint, he/she could be seen from the other viewpoints; while in contained state, such as a person in a vehicle, this person could not be seen from any viewpoints.

In this work, we mainly study the interactions of humans and the developed methods can also be expanded to other objects, e.g., animals.

3.1. Causal And-Or Graph

In this paper, we propose a Causal And-Or Graph (C-AOG) to represent the action-fluent relationship, as illustrated in Fig. 3(a). A C-AOG has two types of nodes: (i) Or-nodes that represent the variations or choices, and (ii) And-nodes that represent the decompositions of the top-level entities. The arrows indicate the causal relations between actions and fluent transitions. For example, a C-AOG can be used to expressively model a series of action-fluent relations.

The C-AOG is capable of representing multiple alternatives for causes of occlusion and potential transitions. There are four levels in our C-AOG: visibility fluents, possible states, state transitions and agent actions. Or nodes represent alternative causes in visibility fluents and state levels; that is, one fluent can have multiple states and one state can have multiple transitions. An event can be decomposed into several atomic actions and represented by an And-node, e.g., an event of a person getting into a vehicle is a composition of four atomic actions: approaching the vehicle, opening the door, entering the vehicle, and closing the door.

Given a video sequence I with length T and camera calibration parameters H, we represent the scene $\mathcal{R}$ as

$$\begin{aligned} \mathcal{R} &= \{O_t : t = 1, 2, ..., T\}, \\ O_t &= \left\{o_t^i : i = 1, 2, ..., N_t\right\}, \end{aligned} \tag{1}$$

where O_t denotes all the objects at time t, and N_t is the size of O_t, i.e., the number of objects at time t. N_t is unknown and will be inferred from observations. Each object o_t^i is represented with its location l_t^i (i.e., bounding boxes in the image) and appearance features ϕ_t^i. To study the visibility fluent of a subject, we further incorporate a state variable s_t^i

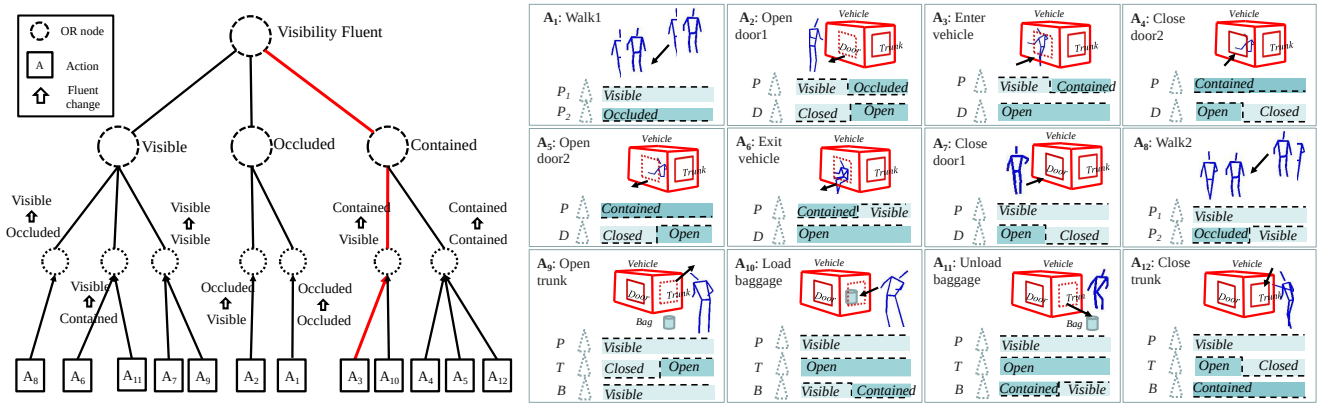

Figure 3. **(a) The proposed Causal And-Or Graph (C-AOG) model for the fluent of visibility.** We use a C-AOG to represent the visibility status of an subject. Each OR node indicates a possible choice and an arrow shows how visibility fluent transits among states. **(b) A series of atomic actions that could possibly cause visibility fluent change.** Each atomic action describes interactions among people and interacting objects. "P", "D", "T", "B" denotes "person", "door", "trunk", "bag", respectively. The dash triangle denotes fluent. The corresponding fluent could be "visible", "occluded" or "contained" for a person; "open", "closed" or "occluded" for a vehicle door or truck. See text for more details.

and an action label a_t^i, that is,

$$o_t^i = \left(l_t^i,\ \phi_t^i,\ s_t^i, a_t^i\right). \tag{2}$$

Thus, the state of a subject is defined as

$$s_t^i \in S = \{\text{ visible, occluded, contained }\}. \tag{3}$$

We define a series of atomic actions $\Omega = \{a_i : i = 1, \ldots, N_a\}$ that might change the visibility status, e.g., walking, opening vehicle door, etc. Fig. 3(b) illustrates a small set of actions Ω covering the most common interactions among people and vehicles.

Our goal is to jointly find subject locations in video frames and estimate their visibility fluents M from the video sequence I. Formally, we have

$$\begin{aligned} M &= \{pg_t : t = 1, 2, \ldots, T\}, \\ pg_t &= \{o_t^i = (l_t^i, \phi_t^i, s_t^i, a_t^i) \,|\, i = 1, 2, ..., N_t\}, \end{aligned} \tag{4}$$

where pg_t can be determined by the optimal causal parse graph at time t.

4. Problem Formulation

According to Bayes' rule, we can solve our joint object tracking and fluent reasoning problem by maximizing a posterior (MAP),

$$\begin{aligned} M^* &= \arg\max_M p(M|I;\theta) \\ &\propto \arg\max_M p(I|M;\theta) \cdot p(M;\theta) \\ &= \arg\max_M \frac{1}{Z} \exp\{-\mathcal{E}(M;\theta) - \mathcal{E}(I|M;\theta)\}. \end{aligned} \tag{5}$$

The **prior** term $\mathcal{E}(M;\theta)$ measures the temporal consistency between successive parse graphs. Assuming G is a Markov Chain structure, we can decompose $\mathcal{E}(M;\theta)$ as

$$\begin{aligned} \mathcal{E}(M;\theta) &= \sum_{t=1}^{T-1} \mathcal{E}(pg_{t+1}|pg_t) \\ &= \sum_{t=1}^{T-1} \sum_{i=1}^{N_t} \Phi(l_{t+1}^i, l_t^i, s_t^i) + \Psi(s_{t+1}^i, s_t^i, a_t^i). \end{aligned} \tag{6}$$

The first term $\Phi(\cdot)$ measures the location displacement. It calculates the transition distance between two successive frames and is defined as:

$$\Phi(l_{t+1}^i, l_t^i, s_t^i) = \begin{cases} \delta(\mathcal{D}_s(l_{t+1}^i, l_t^i) > \tau_s), & s_t^i = \text{Visible}, \\ 1, & s_t^i = \text{Occ, Con}, \end{cases} \tag{7}$$

where $\mathcal{D}_s(\cdot,\cdot)$ is the Euclidean distance between two locations on the 3D ground plane, τ_s is the speed threshold and $\delta(\cdot)$ is an indicator function. The location displacement term measures the motion consistency of object in successive frames.

The second term $\Psi(\cdot)$ measures the state transition energy and is defined as:

$$\Psi(s_{t+1}^i, s_t^i, a_t^i) = -\log p(s_{t+1}^i | s_t^i, a_t^i), \tag{8}$$

where $p(s_{t+1}^i|s_t^i, a_t^i)$ is the action-state transition probability, which can be learned from the training data.

The **likelihood** term $\mathcal{E}(I|M;\theta)$ measures how well each parse graph explains the data, which can be decomposed as

$$\begin{aligned} \mathcal{E}(I|M;\theta) &= \sum_{t=1}^{T} \mathcal{E}(I_t|pg_t) \\ &= \sum_{t=1}^{T} \sum_{i=1}^{N_t} \Upsilon(l_t^i, \phi_t^i, s_t^i) + \Gamma(l_t^i, \phi_t^i, a_t^i), \end{aligned} \tag{9}$$

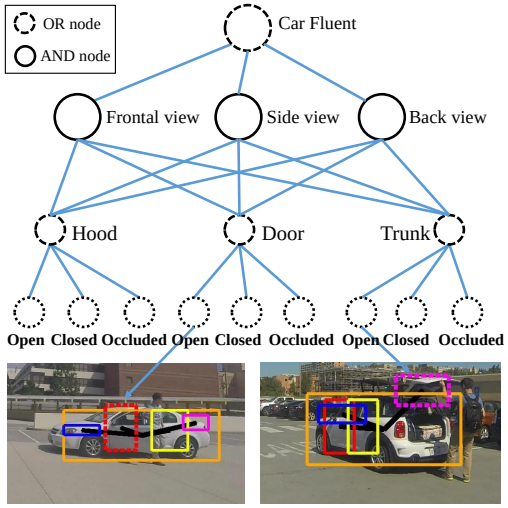

Figure 4. **Illustration of Hierarchical And-Or Graph.** The vehicle is decomposed into different views, semantic parts and fluents. Some detection results are drawn below, with different colored bounding boxes denoting different vehicle parts, solid/dashed boxes denoting state "closed"/"open".

where $\Upsilon(\cdot)$ measures the likelihood between data and object fluents, and $\Gamma(\cdot)$ measures the likelihood between data and object actions. Given each object o_t^i, the energy function $\Upsilon(\cdot)$ is defined as:

$$\Upsilon(l_t^i, \phi_t^i, s_t^i) = \begin{cases} 1 - h_o(l_t^i, \phi_t^i), & s_t^i = \text{Visible}, \\ \sigma(\mathcal{D}_\varsigma(\varsigma_1^i, \varsigma_2^i)), & s_t^i = \text{Occluded}, \\ 1 - h_c(l_t^i, \phi_t^i), & s_t^i = \text{Contained}, \end{cases} \quad (10)$$

where $h_o(\cdot)$ indicates the object detection score, $h_c(\cdot)$ indicates the container (i.e., vehicles) detection score, and $\sigma(\cdot)$ is the sigmoid function. When an object is in either visible or contained state, appearance information can describe the probability of the existence of itself or the object containing it (i.e., container) at this location. When an object is occluded, there is no visual evidence to determine its state. Therefore, we utilize temporal information to generate candidate locations. We employ the SSP algorithm [29] to generate trajectory fragments (i.e., tracklets). The candidate locations are identified as misses in complete object trajectories. The energy is thus defined as the cost of generating a virtual trajectory at this location. We compute this energy by computing the visual discrepancy between a neighboring tracklet ς_1^i before this moment and a neighboring tracklet ς_2^i after this moment. The appearance descriptor of a tracklet is computed as the average pooling of image descriptor over time. If the distance is below a threshold τ_ς, a virtual path is generated to connect these two tracklets using B-spline fitting.

The term $\Gamma(l_t^i, \phi_t^i, a_t^i)$ is defined over the object actions observed from data. In this work, we study the fluents of human and vehicles, that is,

$$\Gamma(l_t^i, \phi_t^i, a_t^i) = \sigma(\mathcal{D}_h(l_t^i, \phi_t^i | a_t^i)) + \sigma(\mathcal{D}_v(l_t^i, \phi_t^i | a_t^i)), \quad (11)$$

where $\sigma(\cdot)$ is the sigmoid function. The definitions of the two data-likelihood terms $\mathcal{D}_h$ and $\mathcal{D}_v$ are introduced in the rest of this section.

A **human** is represented by his/her skeleton, which consists of multiple joints estimated by sequential prediction technology [36]. The feature of each joint is defined as the relative distances of this joint to four saddle points(two shoulders, the center of the body, and the middle between the two hipbones). The relative distances are normalized by dividing the length of head to eliminate the influence of scale. A feature vector ω_t^h concatenating the features of all joints is extracted, which is assumed to follow a Gaussian distribution:

$$\mathcal{D}_h(l_t^i, \phi_t^i | a_t^i) = -\log N(\omega_t^h; \mu_{a_t^i}, \Sigma_{a_t^i}), \quad (12)$$

where $\mu_{a_t^i}$ and $\Sigma_{a_t^i}$ are the mean and the covariance of the action a_t^i respectively, which are obtained from the training data.

A **vehicle** is described with its viewpoint, semantic vehicle parts, and vehicle part fluents. The vehicle fluent is represented by a Hierarchical And-Or Graph, as illustrated in Fig. 4. The feature vector of vehicle fluent ω^v is obtained by computing fluent scores on each vehicle part and concatenating them together. We compute the average pooling feature ϖ_{a_i} for each action a_i over the training data as the vehicle fluent template. Given vehicle fluent ω_t^v computed on image I_t, the distance $\mathcal{D}_v(l_t^i, \phi_t^i | a_t^i)$ is defined as

$$\mathcal{D}_v(l_t^i, \phi_t^i | a_t^i) = \| \omega_t^v - \varpi_{a_t^i} \|_2. \quad (13)$$

5. Inference

We cast the intractable optimization of Eqn. (5) as an Integer Linear Formulation (ILF) in order to derive a scalable and efficient inference algorithm. We use V to denote the locations of vehicles, and E to denote the edges between all possible pairs of nodes, whose time is consecutive and locations are close. The whole transition graph $G = (V, E)$ is shown as Fig. 5. Then the energy function Eqn. (5) can be re-written as:

$$\begin{aligned} f^* &= \arg\max_f \sum_{mn \in E_o} c_{mn} f_{mn}, \\ c_{mn} &= -\Phi(l_n, l_m, s_m) - \Psi(s_n, s_m, a_m) - \Upsilon(l_m, \phi_m, s_m) \\ &\quad - \Gamma(l_m, \phi_m, a_m), \\ s.t. \quad & f_{mn} \in \{0, 1\}, \sum_m f_{mn} \leq 1, \sum_m f_{mn} = \sum_k f_{nk}, \end{aligned} \quad (14)$$

where f_{mn} is the number of object moving from node V_m to node V_n, c_{mn} is the corresponding cost.

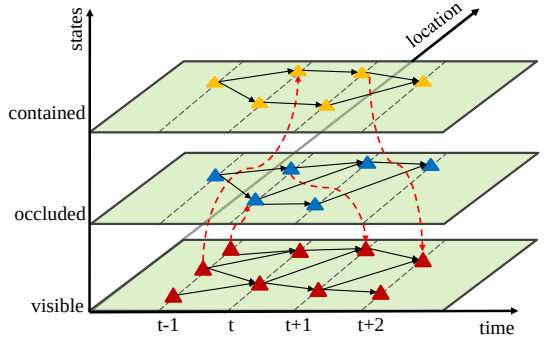

Figure 5. **Transition graph utilized to formulate the integer linear programming.** Each node m has its location l_m, state s_m, and time instant t_m. Black solid arrows indicate the possible transitions in the same state. Red dashed arrows indicate the possible transitions between different states.

Since the subject of interest can only enter a nearby container (e.g., vehicle), to discover the optimal causal parse graph, we need to jointly track the container and the subject of interest. Similar to Eqn. (14), the energy function of container is as follows:

$$\begin{aligned} g^* &= \arg\max_{g} \sum_{mn\in E_c} d_{mn}\, g_{mn}, \\ d_{mn} &= h_c(l_m, \phi_m) - 1, \\ s.t.\ \ g_{mn} &\in \{0,1\},\ \sum_m g_{mn} \le 1,\ \sum_m g_{mn} = \sum_k g_{nk}, \end{aligned} \tag{15}$$

where $h_c(l_m, \phi_m)$ is the container detection score at location l_m. Then we add the contained constrains as:

$$\begin{aligned} &\sum_{mn\in E_c} g_{mn} \ge \sum_{ij\in E_o} f_{ij}, \\ &s.t.\ \ t_n = t_j,\ \|l_n - l_j\|_2 < \tau_c, \end{aligned} \tag{16}$$

where τ_c is the distance threshold. Finally, we combine Eqn. (14)-(16) to obtain objective function for our model:

$$\begin{aligned} f^*, g^* &= \max_{f,g} \sum_{mn\in E_o} c_{mn} f_{mn} + \sum_{ij\in E_c} d_{mn}\, g_{mn}, \\ s.t.\ \ f_{mn} &\in \{0,1\},\ \sum_m f_{mn} \le 1,\ \sum_m f_{mn} = \sum_k f_{nk}, \\ g_{mn} &\in \{0,1\},\ \sum_i g_{mn} \le 1,\ \sum_i g_{mn} = \sum_k g_{nk}, \\ t_n &= t_j,\ \|l_n - l_j\|_2 < \tau_c. \end{aligned} \tag{17}$$

The re-formulated graph still follows a directed acyclic graph (DAG). Thus we can adopt the Dynamic Programming technique to efficiently search for the optimal solution, as illustrated in the Fig. 5.

6. Experiments

We apply the proposed method on two tracking interacting objects datasets and evaluate the improvement in visual tracking brought by the outcomes of visibility status reasoning.

6.1. Implementation Details

We first utilize the Faster R-CNN models [30] trained on the MS COCO dataset to detect involved agents (e.g., person and suitcase). The used network is the VGG-16 net, with score threshold 0.4 and NMS threshold 0.3. The tracklets similarity threshold τ_ς is set as 0.8. The contained distance threshold τ_c is set as the width of container 3 meters. The maximum number of contained objects in a container is set to 5. For appearance descriptors ϕ, we employ the dense sampling ColorNames descriptor [43], which applies square root operator [2] and Bag-of-word encoding to the original ColorNames descriptors. For human skeleton estimation, we use the public implementation of [36]. For vehicle detection and semantic part status estimation, we use the implementation provided by [22] with default parameters mentioned in their paper.

We adopt the widely used CLEAR metrics [19] to measure the performances of tracking methods. It includes four metrics, i.e., Multiple Object Detection Accuracy (MODA), Detection Precision (MODP), Multiple Object Tracking Accuracy (MOTA) and Tracking Precision (MOTP), which take into account three kinds of tracking errors: false positives, false negatives and identity switches. We also report the number of false positives (FP), false negatives (FN), identity switches (IDS) and fragments (Frag). A higher value means better for MODA, MODP, MOTA and MOTP, while a lower value means better for FP, FN, IDS and Frag. If the Intersection-over-Union (IoU) ratio of tracking results to groundtruth is above 0.5, we accept the tracking result as a correct hit.

6.2. Datasets

People-Car dataset [34][1]. This dataset consists of 5 groups of synchronized sequences on a parking lot, recorded from two calibrated bird-view cameras, with length of 300 $\sim$ 5100 frames. In this dataset, there are many instances of people getting in and out of cars. This dataset is challenging for the frequent interactions, light variation and low object resolution.

Tracking Interacting Objects (TIO) dataset. For current popular multiple object tracking datasets (e.g., PETS09 [12], KITTI dataset [15]), most tracked objects are pedestrian and no evident interaction visibility fluent changes. Thus we collect two new scenarios with typical human-object interactions: person, suitcase, and vehicle on several places.

Plaza. We capture 22 video sequences in a plaza that describe people walking around, getting in/out vehicles.

ParkingLot. We capture 15 video sequences in a parking lot that shows vehicles entering/exiting the parking lot, people getting in/out vehicles, people interacting with

[1]This dataset is available at cvlab.epfl.ch/research/surv/interacting-objects

People-Car	Metric	Our-full	Our-1	Our-2	POM [14]	SSP [29]	LP2D [21]	LP3D [21]	KSP-fixed [4]	KSP-free [4]	KSP-seq [4]	TIF-LP [35]	TIF-MIP [35]
Seq.0	FP ↓	0.17	0.34	0.20	0.06	0.04	0.05	0.05	0.46	0.10	0.46	0.07	0.07
	FN ↓	0.08	0.53	0.12	0.47	0.76	0.48	0.53	0.61	0.41	0.61	0.25	0.25
	IDS ↓	0.05	0.07	0.05	-	0.04	0.06	0.06	0.07	0.07	0.07	0.04	0.04
	MODA ↑	**0.71**	0.27	0.63	0.47	0.20	0.47	0.42	-0.07	0.49	-0.07	0.67	0.67
Seq.1	FP ↓	0.21	0.70	0.28	0.98	0.75	0.77	0.75	0.77	0.71	0.75	0.17	0.17
	FN ↓	0.12	0.26	0.14	0.23	0.25	0.21	0.25	0.25	0.25	0.25	0.25	0.25
	IDS ↓	0.04	0.13	0.04	-	0.12	0.17	0.21	0.06	0.12	0.15	0.04	0.04
	MODA ↑	**0.62**	0.09	0.54	-0.21	0.00	0.02	0.00	-0.02	0.04	0.00	0.58	0.58
Seq.2	FP ↓	0.03	0.05	0.04	0.03	0.00	0.03	0.00	0.05	0.00	0.05	0.03	0.03
	FN ↓	0.28	0.58	0.32	0.47	0.59	0.62	0.58	0.72	0.59	0.72	0.47	0.47
	IDS ↓	0.01	0.03	0.02	-	0.01	0.02	0.01	0.03	0.01	0.03	0.01	0.01
	MODA ↑	**0.57**	0.39	0.48	0.50	0.41	0.35	0.42	0.23	0.41	0.23	0.50	0.50
Seq.3	FP ↓	0.18	0.39	0.21	0.59	0.35	0.43	0.27	0.46	0.43	0.43	0.14	0.14
	FN ↓	0.07	0.32	0.10	0.17	0.31	0.23	0.40	0.19	0.23	0.19	0.21	0.21
	IDS ↓	0.06	0.26	0.06	-	0.27	0.34	0.33	0.19	0.25	0.21	0.07	0.05
	MODA ↑	**0.68**	0.35	0.62	0.24	0.34	0.34	0.33	0.35	0.34	0.38	0.65	0.65
Seq.4	FP ↓	0.16	0.27	0.18	0.40	0.19	0.26	0.13	0.32	0.25	0.31	0.08	0.07
	FN ↓	0.10	0.18	0.13	0.15	0.19	0.16	0.18	0.17	0.17	0.16	0.16	0.15
	IDS ↓	0.05	0.15	0.05	-	0.14	0.13	0.15	0.12	0.12	0.11	0.04	0.04
	MODA ↑	**0.82**	0.59	0.73	0.45	0.62	0.58	0.69	0.51	0.58	0.53	0.76	0.78

Table 1. **Quantitative results and comparisons** of false positive (FP) rate, false negative (FN) rate and identity switches (IDS) rate **on People-Car Dataset**. The best scores are marked in **bold**.

Plaza	MOTA ↑	MOTP ↑	FP ↓	FN ↓	IDS ↓	Frag ↓
Our-full	**46.0%**	76.4%	99	501	5	8
Our-1	31.9%	75.1%	40	643	29	36
Our-2	32.5%	75.3%	75	605	25	30
MHT_D [20]	34.3%	73.8%	56	661	15	18
MDP [38]	32.9%	73.2%	24	656	9	7
DCEM [26]	32.3%	**76.5%**	2	675	2	2
SSP [29]	31.7%	72.1%	19	678	21	25
DCO [1]	29.5%	76.4%	22	673	6	2
JPDA_m [18]	13.5%	72.2%	163	673	6	3
ParkingLot	MOTA ↑	MOTP ↑	FP ↓	FN ↓	IDS ↓	Frag ↓
Our-full	**38.6%**	**78.6%**	418	1954	6	5
Our-1	28.7%	78.4%	451	2269	15	17
Our-2	28.9%	78.4%	544	2203	14	16
MDP [38]	30.1%	76.4%	397	2296	26	22
DCEM [26]	29.4%	77.5%	383	2346	16	15
SSP [29]	28.9%	75.0%	416	2337	12	14
MHT_D [20]	25.6%	75.7%	720	2170	15	12
DCO [1]	24.3%	78.1%	536	2367	38	10
JPDA_m [18]	12.3%	74.2%	1173	2263	28	17

Table 2. **Quantitative results and comparisons** of false positive (FP), false negative (FN), identity switches (IDS), and fragments (Frag) on **TIO dataset**. The best scores are marked in **bold**.

trunk/suitcase.

All video sequences are captured by a GoPro camera, with frame rate 30fps and resolution 1920×1080. We use the standard chessboard and Matlab camera calibration toolbox to obtain camera parameters. The total number of frames of TIO dataset is more than 30K. There exist severe occlusions and large scale changes, making this dataset very challenging for traditional tracking methods.

Beside the above testing data, we collect another set of video clips for training. To avoid over-fitting, we set up different camera positions, different people and vehicles from the testing settings. The training data consists of 380 video clips covering 9 events: *walking, opening vehicle door, entering vehicle, exiting vehicle, closing vehicle door, opening vehicle trunk, loading baggage, unloading baggage, closing vehicle trunk*. Each action category contains 42 video clips on average.

Both the datasets and short video clips are annotated with bounding boxes for people, suitcases, vehicles, and visibility fluents of people and suitcases. The types of status are "visible", "occluded", and "contained". We utilize VATIC [32] to annotate the videos.

6.3. Results and Comparisons

For People-Car dataset, we compare our proposed method with 5 baseline methods and their variants: successive shortest path algorithm (SSP) [29], K-Shortest Paths Algorithm (KSP-fixed, KSP-free, KSP-seq) [4], Probability Occupancy Map (POM) [14], Linear Programming (LP2D, LP3D) [21], and Tracklet-Based Intertwined Flows (TIF-IP, TIF-MIP) [35]. We refer the reader to [35] for more details about the method variants. The quantitative results are reported in Table 1. From the results, we can observe that the proposed method obtains better performance than the baseline methods.

For TIO dataset, we compare the proposed method with 6 state-of-the-arts: successive shortest path algorithm (SSP) [29], multiple hypothesis tracking with distinctive appearance model (MHT_D) [20], Markov Decision Processes with Reinforcement Learning (MDP) [38], Discrete-Continuous Energy Minimization (DCEM) [26], Discrete-continuous optimization (DCO) [1] and Joint Probabilistic Data Association (JPDA_m) [18]. We use the public implementations of these methods.

We report quantitative results and comparisons in Table 2

Figure 6. **Sampled qualitative results of our proposed method on TIO dataset and People-Car dataset.** Each color represents an object. The solid bounding box means the visible object. The dash bounding box denotes the object is contained by other scene entities. Best viewed in color and zoom in.

for TIO dataset. From the results, we can observe that our method obtains superior performance to the other methods on most metrics. It validates that the proposed method can not only track visible objects correctly, but also reason locations for occluded or contained objects. The alternative methods do not work well mainly due to lack of the ability to track objects under long-term occlusion or containment in other objects.

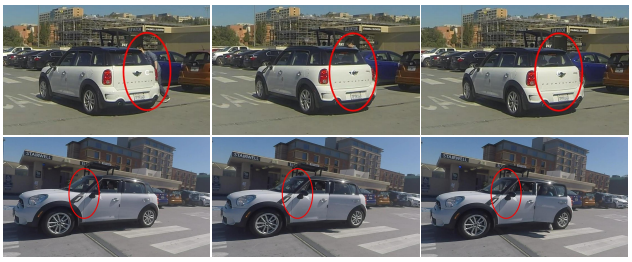

Figure 7. **Sampled failure cases**. When people stay behind vehicles, it is hard to determine whether or not they are interacting with the vehicle, e.g., entering, exiting.

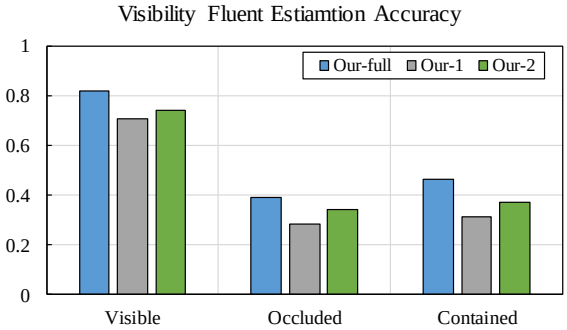

Figure 8. **Visibility fluent estimation** results on TIO dataset.

We set up three baselines to analyze the effectiveness of different components in the proposed method:

- **Our-1**: no likelihood term and only prior term is used.
- **Our-2**: only human data-likelihood term and prior term are used.
- **Our-full**: all terms are used, including prior terms, human and vehicle data-likelihood terms.

Based on comparisons of Our-1, Our-2 and Our-full, we can also conclude that each type of fluent plays its role in improving the final tracking results. Some qualitative results are displayed in Fig. 6.

We further report fluent estimation results on TIO-Plaza sequences and TIO-ParkingLot sequences in Fig. 8. From the results, we can see that our method can successfully reason the visibility status of subjects. Note that the precision of containment estimation is not high, since some people get in/out the vehicle from the opposite side towards the camera, as shown in Fig. 7. Under such situation, there are barely any image evidence to reason the object status and multi-view setting might be a better way to reduce the ambiguities.

7. Conclusion

In this paper, we propose a Causal And-Or Graph (C-AOG) model to represent the causal-effect relations between object visibility fluents and various human interactions. By jointly modeling short-term occlusions and long-term occlusions, our method can explicitly reason the visibility of subjects as well as their locations in the videos. Our method clearly outperforms the alternative methods in complicated scenarios with frequent object interactions. In this work, we focus on the human-interactions as a running-case of the proposed technique, and we will explore the extension of our method to other types of objects (e.g., animal, drones) in the future.

References

[1] A. Andriyenko, K. Schindler, and S. Roth. Discrete-continuous optimization for multi-target tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 7

[2] R. Arandjelovic and A. Zisserman. Three things everyone should know to improve object retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 6

[3] T. Baumgartner, D. Mitzel, and B. Leibe. Tracking people and their objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2

[4] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(9):1806–1819, 2011. 2, 7

[5] W. Choi and S. Savarese. A unified framework for multi-target tracking and collective activity recognition. In *European Conference on Computer Vision*, 2012. 2

[6] A. Dehghan, S. Assari, and M. Shah. Gmmcp-tracker:globally optimal generalized maximum multi clique problem for multiple object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[7] A. Dehghan, Y. Tian, P. Torr, and M. Shah. Target identity-aware network flow for online multiple target tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2

[8] X. Dong, J. Shen, W. Wang, Y. Liu, and L. Shao. Hyperparameter optimization for tracking with continuous deep q-learning. In *CVPR*, 2018. 2

[9] X. Dong, J. Shen, D. Yu, W. Wang, J. Liu, and H. Huang. Occlusion-aware real-time object tracking. *IEEE Transactions on Multimedia*, 19(4):763–771, 2017. 2

[10] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu. Learning pose grammar to encode human body configuration for 3D pose estimation. In *AAAI*, 2018. 3

[11] P. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 2

[12] J. Ferryman and A. Shahrokni. Pets2009: Dataset and challenge. In *IEEE International Workshop on Performance Evaluation of Tracking and Surveillance*, 2009. 6

[13] A. Fire and S.-C. Zhu. Learning perceptual causality from video. *ACM Transactions on Intelligent Systems and Technology*, 7(2), 2016. 3

[14] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):267–282, 2008. 7

[15] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 6

[16] T. Griffiths and J. Tenenbaum. Structure and strength in causal induction. *Cognitive Psychology*, 51(4):334–384, 2005. 3

[17] T. Griffiths and J. Tenenbaum. Two proposals for causal grammars. *Causal learning: Psychology, philosophy, and computation*, pages 323–345, 2007. 3

[18] S. Hamid Rezatofighi, A. Milan, Z. Zhang, Q. Shi, A. Dick, and I. Reid. Joint probabilistic data association revisited. In *IEEE International Conference on Computer Vision*, 2015. 7

[19] R. Kasturi, D. Goldgof, P. Soundararajan, V. Manohar, J. Garofolo, R. Bowers, M. Boonstra, V. Korzhova, and J. Zhang. Framework for performance evaluation of face, text, and vehicle detection and tracking in video: Data, metrics, and protocol. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):319–336, 2009. 6

[20] C. Kim, F. Li, A. Ciptadi, and J. M. Rehg. Multiple hypothesis tracking revisited. In *IEEE International Conference on Computer Vision*, 2015. 7

[21] L. Leal-Taixé, M. Fenzi, A. Kuznetsova, B. Rosenhahn, and S. Savarese. Learning an image-based motion context for multiple people tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 7

[22] B. Li, T. Wu, C. Xiong, and S.-C. Zhu. Recognizing car fluents from video. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 6

[23] W. Liang, Y. Zhao, Y. Zhu, and S.-C. Zhu. What are where: Inferring containment relations from videos. In *International Joint Conference on Artificial Intelligence*, 2016. 3

[24] J. Liu, P. Carr, R. T. Collins, and Y. Liu. Tracking sports players with context-conditioned motion models. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2013. 2

[25] A. Maksai, X. Wang, and P. Fua. What players do with the ball: A physically constrained interaction modeling. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[26] A. Milan, K. Schindler, and S. Roth. Multi-target tracking by discrete-continuous energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(10):2054–2068, 2016. 7

[27] E. T. Mueller. *Commonsense reasoning: An event calculus based approach*. Morgan Kaufmann, 2014. 2

[28] J. Pearl. *Causality: Models, reasoning and inference*. Cambridge University Press, 2009. 3

[29] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 5, 7

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Conference on Neural Information Processing Systems*, 2015. 2, 6

[31] T. Shu, S. Todorovic, and S.-C. Zhu. Cern: Confidence-energy recurrent network for group activity recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 2

[32] C. Vondrick, D. Patterson, and D. Ramanan. Efficiently scaling up crowdsourced video annotation. *International Journal of Computer Vision*, 101(1):184–204, 2013. 7

[33] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In *CVPR*, 2018. 3

[34] X. Wang, E. Turetken, F. Fleuret, and P. Fua. Tracking interacting objects optimally using integer programming. In *European Conference on Computer Vision*, 2014. 2, 6

[35] X. Wang, E. Turetken, F. Fleuret, and P. Fua. Tracking interacting objects using intertwined flows. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 38(11):2312–2326, 2016. 2, 7

[36] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 5, 6

[37] L. Wen, W. Li, J. Yan, and Z. Lei. Multiple target tracking based on undirected hierarchical relation hypergraph. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014. 2

[38] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *IEEE International Conference on Computer Vision*, 2015. 7

[39] Y. Xu, L. Lin, W.-S. Zheng, and X. Liu. Human re-identification by matching compositional template with cluster sampling. In *CVPR*, 2016. 3

[40] Y. Xu, X. Liu, Y. Liu, and S.-C. Zhu. Multi-view people tracking via hierarchical trajectory composition. In *CVPR*, 2016. 2

[41] Y. Xu, X. Liu, L. Qin, and S.-C. Zhu. Cross-view people tracking by scene-centered spatio-temporal parsing. In *AAAI*, 2017. 2

[42] S.-I. Yu, D. Meng, W. Zuo, and A. Hauptmann. The solution path algorithm for identity-aware multi-object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2

[43] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian. Scalable person re-identification: A benchmark. In *IEEE International Conference on Computer Vision*, 2015. 6